International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

June 2021 | Issue 47 | Volume 13 | Number 2 | Pages 138-143

DYNAMIC VISUALIZATION OF A STATIC IMAGE WITH AN ACTIONABLE QR CODE

V. Jain [1] Y. Jain [1] M. Agarwal [1] H. Dhingra [1] D. Saini [1] M.C. Taplamacioglu [2] M. Saka [2]

1. Bharati Vidyapeeth's College of Engineering, New Delhi, India, vanita.jain@bharatividyapeeth.edu, yugantar.jain@icloud.com, mahima6161@gmail.com, hardikdhingra@gmail.com, dsaini77@gmail.com
2. Electrical and Electronics Engineering Department, Gazi University, Ankara, Turkey, taplam@gazi.edu.tr, msaka@gazi.edu.tr

Abstract- An image is generally static in nature which is readable by the users but not actionable. A file type such as a PDF on the other hand is actionable with support for hyperlinks for web links, phone numbers, email addresses and more. In this paper, it is proposed as the technology that can visualize a static image and make it dynamic and directly actionable starting with a QR code. In recent times, a lot of information is shared through images, in the form of posters for an event, digital cards and more. The dynamic visualizer for the static image will help make these images user interactable for added convenience, safety and efficiency.

Keywords: QR Code, Detection Actionable, Dynamic Image.

1. INTRODUCTION

A QR code (Quick Response code) is a two-dimensional barcode using the ISO/IEC 18004:2006 (now revised in 18004:2015) standard [1]. A QR code is generated after certain protocols are followed. The same protocols are utilised for its decoding. The generation of the QR code is a straightforward process. The basic architecture of the QR code will be summarised in the coming sections of this paper. In this study, the dynamic visualization (for a QR code) of a static image with specific actions for different data types including text, web links, email addresses, and phone numbers are developed. The visualization is a four-step process as shown in Figure 1, it involves:

1. Detection and decoding of QR code,
2. Classification of decoded data,
3. Actionable button overlay on frame of QR code,
4. Platform specific actions.

For the detection of the QR code in a static image, the Apple Developer Core Image framework [2] is used. This framework tackles the parameters of QR code Detection- The detection or localisation of QR codes and the pre-processing of QR codes. Next, after detection of the QR code and decoding of its data, we classify that data into normal text, phone numbers, email addresses, and web links using the NSDataDetector API [3].

An actionable button is placed as on overlay over the QR code, this is done using the coordinates of its frame/bounds detected in the first step. When a user presses this button, a specific action (platform dependent) for the particular data Figure 1. Flow diagram to make QR code actionable type is triggered. In this implementation, it is developed actions for normal text, web links, email addresses, and phone numbers; these are showing text, opening web link in Safari browser, composing mail in mail app, and making phone call respectively on the iOS platform.

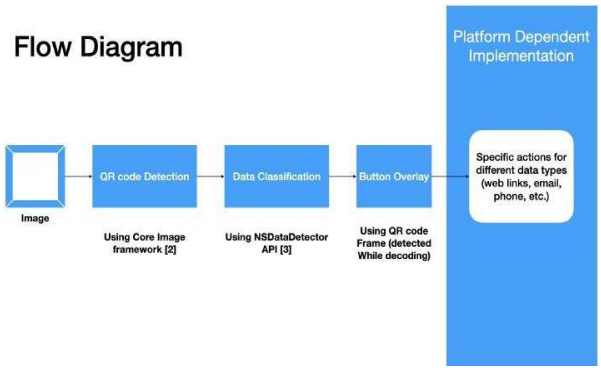


Figure 1. Flow diagram to make QR code actionable

2. RELATED WORK

The first step in this work is to detect and decode the data from a QR code in an image. It is done an extensive study on this and analysed different methods and algorithms to detect QR codes and compiled them in the previous review paper [4]. During this study, it is found that ZXing [5] is a very popular open-source library for QR code decoding and has been used by many authors to compare their work and its performance. Here it is analysed the different QR code detection and decoding techniques from 20+ different papers. A few algorithms are found to be demonstrably superior, for example, the Hansen et al. 2017 [6] that used YOLO with the best CPU performance as reported by its authors, the K Suran 2013 [7] for best accuracy in 2D rotation, and M Li et al. 2017 [8] for best accuracy in 3D rotation.

Hansen et al. 2017 [6] adopted the deep learning based detector of YOLO (You Only Look Once) [7] for detection of 1D and 2D barcodes. They are also able to achieve image pre-processing in terms of rotation using the same model and are able to increase the decoding performance based on tests performed using the ZXing [8] and Zbar [8] open-source library for QR code decoding. This work shows one of the simplest and perhaps the most promising technique for barcode localization and rotation for everyday uses.

The authors use the Soros [9] and Dubeska [10] barcode datasets for testing and achieve a laudable 1.0 detection rate with bounding boxes for the algorithm and a 95% average accuracy. Here, it is proposed research papers for QR code image correction and preprocessing, too. G Soroz et al. 2014 discussing about removal of uniform blur with up to 10x runtime improvements [9], B. Wang et al. 2019 [11] removes motion blur with one of the highest avg PSNR value using GAN (Generative Adversative Network) and Y. He et al. 2019 [12] with one of the highest detection rate in uneven illumination using adaptive local binarization. It was easy to realize that there is no one best algorithm for all cases, and that different methodologies are optimal for different use cases. In the review paper, Singh [13] presents the common workflow for image preprocessing in special regard to the angular distortion and is shown in Figure 2.

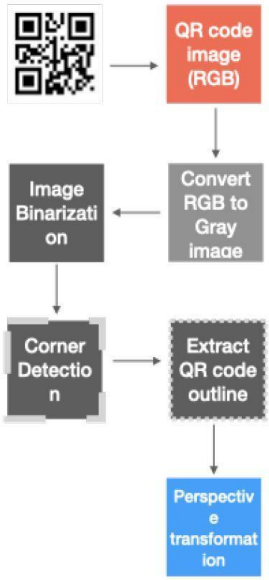


Figure 2. QR code image pre-processing flow diagram [13]

For classification of text into data types such as normal text, web links, email addresses, phone numbers, etc, C. Madan Kumar, M. Brindha have done extensive work and developed algorithms for classification of text into a wide range of data types in their paper Text Extraction from Business Cards and Classification of Extracted Text Into Predefined Classes [14]. Here, the authors use keyword checking as an initial check for a lot of classes. For example: checking "www" prefix for web links, "@" for email addresses, numerical digits for phone numbers and so on.

They have used the package 'libpostal' for parsing of string for the above classifications and Tesseract OCR for text extraction. The authors do text classification for 15 different data points and thework in this paper achieves greater than 95% accuracy for each class. Alper Kursat Uysal and Serkan Gunal evaluate classification accuracy, text domain, text language, and dimension reduction in their paper the impact of preprocessing on text classification [15]. All combinations of pre- processing tasks for text classification were examined in this paper using two languages- Turkish and English on two different domains email and news. They gave experimental analysis on datasets to point out appropriate combinations of preprocessing tasks for massive improvement on classification accuracy. Thien Hai Nguyen and Kiyoaki Shirai use features- Title Bi-gram and Title SigNoun to determine the research topics of a given technical paper in their paper Text Classification of Technical Papers Based on Text Segmentation [16]. For text classification, title, abstract, introduction and conclusion are used as segments.

The specific actions performed for the particular data types on tapping of the QR code are comparable to the way hyperlinks for different data types work in a Portable Document Format (PDF) document.

3. TECHNOLOGY

As mentioned earlier, in the recent study is dynamically visualized a static image for the contained QR code through a series of steps. Now, it is going to describe each of them in detail here.

3.1. QR Code Detection

A QR (Quick Response) code is a 2D barcode that was invented by Denso Wave [17] in 1994, shown in Figure 3.

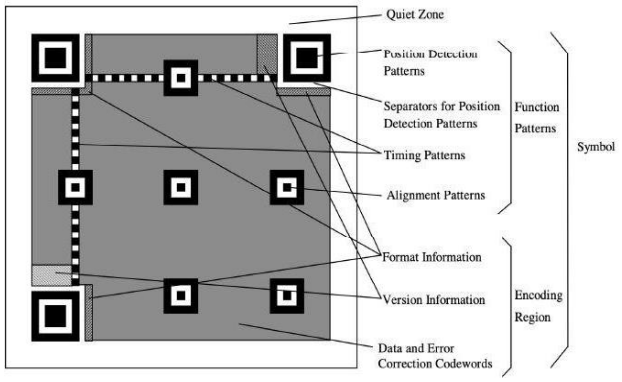


Figure 3. Structure of QR code symbol, standard defined in [1]

The structure of QR code consists of Position detection patterns to indicate direction in which the code is printed, alignment pattern which helps in orientation, timing patterns to determine the size of data matrix, version and format information for error tolerance, error correction code to hold the data and quiet zone for the scanner to distinguish the QR Code from its surroundings [18]. To detect and decode the data of the QR code from an image, it is used the Apple Developer's Core Image framework [2]. Using Core Image, it is possible to detected a QR code and get the encoded message contained in it along with

coordinates of its frame. This information is used to enabler the whole system including data classification and actions and button overlay for dynamic visualization.

Our QR code detection algorithm works by first getting the static image and using its data to build a Core Image object. This Core Image object is essential since we use the Core Image framework's power to decode the QR code. Next, we declare a dictionary containing the options for QR code detection. This dictionary uses a String to Any mapping in Swift and we set the option for CIDetectorAccuracy to high here. Now, it is instantiated a new CIDetector object with a default CIContext and options and specify its type to CIDetectorTypeQRCode. The type parameter specifies the type of data object to be decoded, this in the suggested case is a 2D QR Code. Next, it is checked for any rotation in the QR code and fix the orientation of the source image. This is done by updating the options dictionary we declared earlier for the key CIDetectorImageOrientation. If no rotation is detected, it is set the value to 1. Now, for the final steps, the features of the QR Code through our detector object and return those to the program is simply got. This whole flow is shown visually in Figure 4.

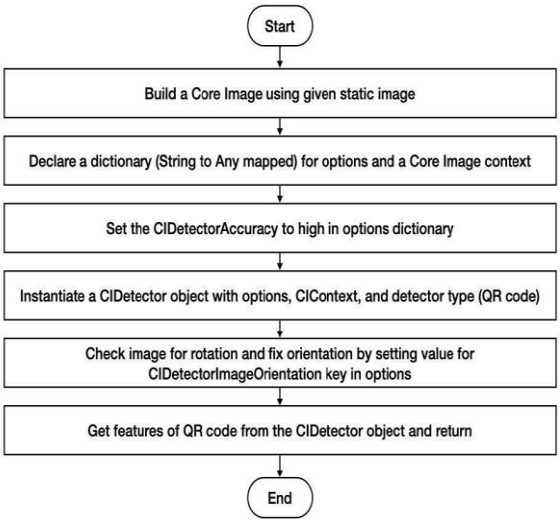


Figure 4. The suggested QR detection algorithm flow

3.2. Data Classification

For the classification of the data/text decoded from the QR code, it is used the Apple Developer NSDataDetector API [3]. In addition, to classify the data into the following types (which then form the basis for specialized actions) is used:

- Normal Text
- Web Links
- Email Addresses
- Phone Numbers

This API has helped to quickly and accurately classify all the data points from the used dataset of QR codes into suitable data types. To detect the data type of the message decoded from the QR code, first create an NSDataDetector object while specifying the types to be checked for. Next, it is used the detector object to get matches for the data types as mentioned in the data.

It is obtained a match's sequence and each match contains data and meta-data through properties) about the type. Next, it is switched through the different types of data that it can have and define the action to be performed for each as described below.

3.3. Actions for Specific Data Types

Specific actions are triggered for different data types when a user interacts with the static image by tapping on the QR code. Table 1 gives a comprehensive overlook of the data types handled and the actions for them.

Table 1. Data types with their specific actions

Data Type	Action Performed
Normal Text	Text Shown
Web Link	Link opened in Safari browser
Email Address	Mail app opened with pre-filled email address
Phone Number	Call initiated

It shall be noted that the actions are platform dependent as well as their implementation. Here it is used the iOS platform for proposed implementation. To achieve opening of the browser, the mail app, and calling on the user device, deep linking technology is used. For that the 'open(:options:completionHandler:)' API of UIKit is used [18].

4. RESULTS

To test the proposed work for dynamic visualization of a static image with a QR code, it is used the 'QR Code noise images' dataset [20] by Aurio Pinto. This dataset contains a lot of hard to decode blurred images of QR code, this pushes the proposed QR code detection method to its limits and checks for real life scenarios. Additionally, the data contained in the QR codes in this dataset are wide-ranging, from normal text to web links. This helps us to test classification of decoded message into different data types too. While the dataset does not contain phone numbers and email addresses, additional testing with custom data has shown promises results for them too.

It is achieved that an accuracy of 80.64% for QR code detection with our method (Core Image). This is much higher than the 76.34% accuracy achieved by ZXing (a popular opensource library for 2D and 3D Barcode detection) on the same dataset that we've used. This is shown in Fig. 5. By using Core Image for QR code detection, we have been able to achieve high accuracy and decode even some of the most complex QR codes. For example, the QR code shown in Fig. 6 is successfully decoded by the suggested method whereas the ZXing reader failed to do it. The used dataset for QR codes contained two types of data: normal text and web links. The used accurately classified as the data in each case using NSDataDetector, and a sample of the data along with classification results are given in Table 2. The average time used to detect the QR codes was 0.00775636 seconds. The system used is packed with the ARM-based Apple M1 chip containing 8 CPU cores with max clock rate of 3.2 GHz, 8 GB 4266 MHz SDRAM and an NVME SSD storage [21].

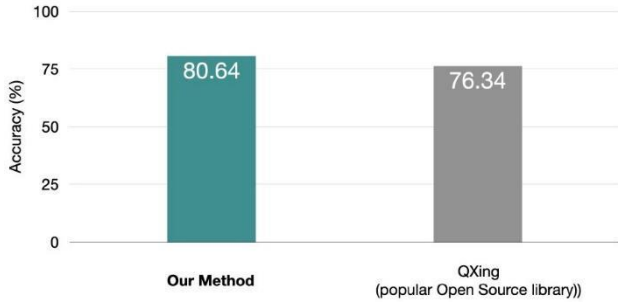


Figure 5. Accuracy comparison for QR code detection

Figure 6. Complex QR code decoded by proposed method while ZXing failed

Table 2. Sample data with classification results

Sample QR Code	Decoded Data	Classification Results
	https://DoItWell.app/	Web Link
	Scanner	Normal Text
	https://qrcode-monkey.com	Web Link
	goo.gl/m6sseN	Web Link

The average time taken by the method is compared with other methods and is shown in Figure 7.

Figure 7. Performance comparison of QR code detection

It can be seen that Hansen et al., who used deep learning based detector of YOLO to build one of the most promising techniques to detect QR codes takes almost double the time than our method. In the next part, the results of classification of the data in the QR code detected will explained.

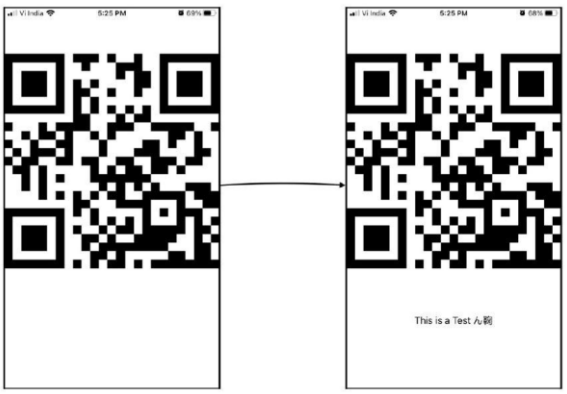


Figure 8. Classification of QR code data as text

In Figure 8, the data in the QRcode is correctly classified as text and shown below the image itself. This does not require redirection of any sort.

Figure 9. Classification of QR code data as web link and redirection to a browser

In Figure 9, the QR code is detected and the data is successfully classified as a web link. It is then redirected to a web browser to open it for further actions.

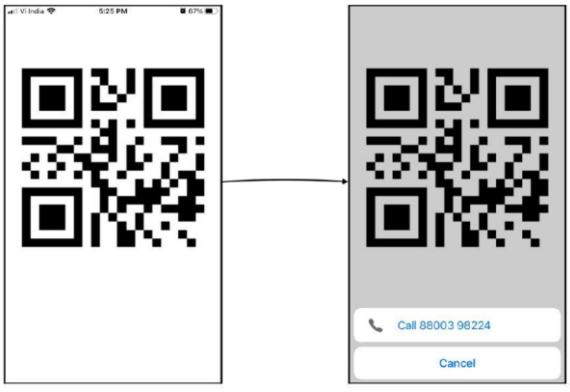


Figure 10. Classification of data as a phone number and actionable button placed

Figure 10 shows the data classified as a phone number and an actionable button from the phone application appearing to take further action which in this case is calling the number.

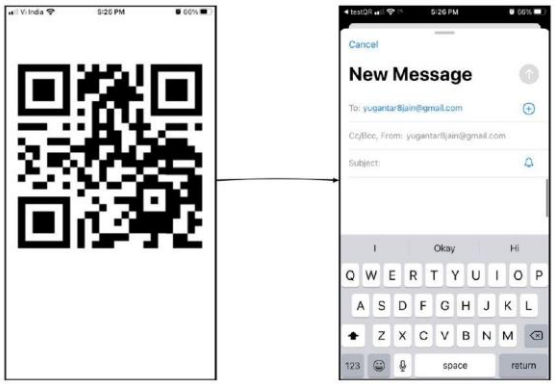


Figure 11. Classification of data as email address & redirection to the mail app

Figure 11 shows the successful classification of data as an email address and automatically redirecting to the mailing application of the phone for further actions.

5. CONCLUSIONS

In this paper, it is shown that an implementation to dynamically visualize a static image for a QR code in it and make it user interact-able and actionable. This has been achieved by state-of-the-art algorithm for QR code detection and decoding, data classification, and platform specific actions implementation using deep linking. Documents such as PDF have supported hyperlinks and the ability to directly interact with elements like web links, emails, and phone numbers for a long time.

The main goal here is to be able to achieve the same level of functionality in a normal static image and go even beyond that by supporting dynamic visualization of a QR code too. With digital transformation of information and a lot of it happening through images of different formats, the technology described is a really significant utility. Moreover, the proposed dynamic visualizer is independent of data types, i.e., it supports all the commonly used different types of images. In future, it is planning to build on the foundations of data specific actions that the developed and implement the complete dynamic visualizer for a static image where the user can interact with the all the hyperlinks along with the QR code.

REFERENCES

[1] "Automatic identification and data capture techniques - QR Code bar code symbology specification", ISO/IEC 18004, Switzerland, 2015, https://webstore.ansi.org/ Previews/PREVIEW ISO+IEC+18004- 2015.pdf.

[2] Core image framework, apple developer, https://developer.apple.com/documentation/coreimage.

[3] NSDataDetector, apple developer, https://developer. apple.com/documentation/foundation/nsdatadetector.

[4] V. Jain, Y. Jain, H. Dhingra, D. Saini, M.C. Taplamacioglu, M. Saka, "A Systematic Literature Review on QR Code Detection and Pre-processing", International Journal on Technical and Physical Problems of Engineering, Iss. 46, Vol. 13, No. 1, pp. 111-119, March 2021.

[5] ZXing, QR code decoding library, https://github.com /zxing/zxing.

[6] D.K. Hansen, K. Nasrollahi, C.B. Rasmussen, T.B. Moeslund, "Real-Time Barcode Detection and Classification Using Deep Learning", 9th International Joint Conference on Computational Intelligence, pp. 321-327, 2017.

[7] J. Redmon, A. Farhadi, "YOLO9000: Better, faster, stronger", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517-6525, 2017.

[8] ZBar, barcode reading library, https://github.com /ZBar/ZBar.

[9] G. Soros, C. Florkemeier, "Blur-Resistant Joint 1D and 2D Barcode Localization for Smartphones", Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia, pp. 1-8, 2013.

[10] M. Dubska, A. Herout, J. Havel, "Real-Time Precise Detection of Regular Grids and Matrix Codes", Journal of Real-Time Image Processing, Vol. 11, pp. 193-200, 2016.

[11] B. Wang, J. Xu, J. Zhang, G. Li, X. Wang, "Motion Deblur of QR Code Based on Generative Adversative Network", 2nd International Conference on Algorithms, Computing and Artificial Intelligence, pp. 166-170, 2019.

[12] Y. He, Y. Yang, "An Improved Sauvola Approach on QR Code Image Binarization", IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT), pp. 6-10, 2019.

[13] A. Singh, P. Singh, "A review: QR Codes and Its Image Pre-processing Method", International Journal of Science, Engineering and Technology Research, Vol. 5, Iss. 6, pp. 1955-1960, 2016.

[14] C.M. Kumar, M. Brindha, "Text extraction from business cards and classification of extracted text into predefined classes", International Conference on Computational Intelligence & IoT (ICCIIoT), pp. 595-602, 2018.

[15] A.K. Uysal, S. Gunal, "The Impact of Preprocessing on Text Classification", Information Processing & Management, Vol. 50, Iss. 1, pp. 104-112, 2014.

[16] T.H. Nguyen, K. Shirai, "Text Classification of Technical Papers Based on Text Segmentation", 18th International Conference on Applications of Natural Language to Information Systems, pp. 278-284, 2013.

[17] Denso Inc., "QR code webpage," "https://www. qrcode.com/en/index.html", 2019.

[18] J. Qianyu, "Exploring the Concept of QR Code and the Benefits of Using QR Code for Companies", Bachelor's Thesis, School of Business and Culture Degree Programme in Business Information Technology, 2014.

[19] "Open URL API", UIKit, Apple developer, https://developer.apple.com/documentation/uikit/uiapplic ation/1648685-open.

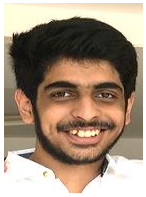
[20] "QR Code noisy images", dataset by Aurio Pinto, https://developer.apple.com/documentation/uikit/uiapplic ation/1648685-open.

[21] Apple M1 Chip: Specs and performance by Michelle Ehrhardt, https://www.tomshardware.com/news/Apple-M1-Chip-Everything- We- Know.

BIOGRAPHIES

Vanita Jain received B.E. in Electrical Engineering in 1984, M. Tech in Controls in 1990 and Ph.D. degree from V.J.T.I., Mumbai, India. She is the first woman to have completed Ph.D. in Technology from Mumbai University. She is having more than 30 years of teaching experience, and taught students at both UG and PG level. Currently, she is working as Professor and Head of Information Technology Department at Bharati Vidyapeeth's College of Engineering, New Delhi, India. Her field of interest includes soft computing, control, optimization techniques and system engineering and is actively involved in the research in these areas.

Yugantar Jain is pursuing a Bachelor's degree in Information Technology and will be graduating in Summer 2021. He is currently a student in the Department of Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, India. He is a former Google Summer of Code student and a recipient of Apple WWDC20 SSC award. His research interests and subjects are QR codes, dynamic images, text extraction, text classification, and data detection.



Hardik Dhingra is pursuing a Bachelor's degree in Information Technology and will be graduating in Summer 2021. He is a student in the Department of Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, India. His research interests and subjects are QR codes, dynamic images, text extraction, text classification and data detection.

Mahima Agarwal is pursuing a Bachelor's degree in Information Technology from Bharati Vidyapeeth's College of Engineering, New Delhi and will be graduating in Summer 2021. She is currently training at Accenture to become a full time Application Development Associate. Her research interests and subjects are QR codes, dynamic images, text extraction, text classification, and data detection.

Dharmender Saini graduated in Computer Science and Engineering from Technological Institute of Textile & Sciences, Bhiwani (A Birla Institute), India. He received the degree of M. Tech. in Computer Science and Engineering from Department of CSE, Guru Jhambeshawar University, Hissar, India. He received the degree of Ph.D. in Computer Science and Engineering from Jamia Millia Islamia, Delhi, India. He has been a Professor of Computer Science & Engineering since 2013. He has also been serving as a Principal of Bharati Vidyapeeth's college of Engineering, India since Jan, 2014. His research interests and subjects are cryptography, automata theory, and foundation of computer science.

M. Cengiz Taplamacioglu graduated from Department of Electrical and Electronics Engineering, Gazi University (Ankara, Turkey). He received the degrees of M.Sc. in Industrial Engineering from Gazi University and in Electrical and Electronics Engineering from Middle East Technical University (Ankara, Turkey) and received the degree of Ph.D. in Electrical, Electronics and System Engineering from University of Wales (Cardiff, UK). He has been a Professor of the Electrical and Electronics Engineering since 2000. His research interests and subjects are power system control, high voltage engineering, corona discharge and modeling, electrical field computation, measurement and modeling techniques, optical HV measurement techniques, lighting techniques, renewable energy systems and smart grid applications.

Mustafa Saka received the B.Sc. degree in Electrical and Electronics Engineering from Pamukkale University (Denizli, Turkey) in 2012 and M.Sc. degree in Electrical and Electronics Engineering from Gazi University (Ankara, Turkey) in 2017. He is a research assistant Electrical and Electronics Engineering Department of Gazi University. His research interests are in electrical installations, power system analysis, control systems and optimization.