www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 06, June 2025, pp : 468-475

e-ISSN : 2583-1062
Impact Factor : 7.001

QR DETECTION USING MATLAB

Ghatadi Suraji Deekshitha[1], Barad Sushith[2], Dhyavarishetty Srinivas[3], Mr.Lingampally Shiv Prasad[4], Mrs.Lomte Neetha[5]

[1,2,3]Student Department Of Electronics And Communication Engineering, Kshatriya College Of Engineering, Armoor, Dist: Nizamabad, Telangana, India

[4]Asst.Prof Department Of Electronics And Communication Engineering, Kshatriya College Of Engineering, Armoor, Dist: Nizamabad, Telangana, India

[5]Guide & Asst.Prof Department Of Electronics And Communication Engineering, Kshatriya College Of Engineering, Armoor, Dist: Nizamabad, Telangana, India

ABSTRACT

This project focuses on the detection and decoding of QR codes using MATLAB, a high-level programming environment well-suited for image processing tasks. The objective is to develop an efficient algorithm capable of accurately identifying QR codes in digital images, extracting their encoded data, and decoding it to retrieve the original information. The approach leverages MATLAB's built-in image processing toolbox to preprocess images, detect QR code patterns through edge detection and morphological operations, and apply geometric transformations for alignment. The system then decodes the QR code using error correction techniques to ensure data integrity even in noisy or partially damaged images. The implemented method demonstrates robustness and reliability across various sample images, highlighting MATLAB's potential for developing practical computer vision applications related to QR code recognition.

1. INTRODUCTION

QR (Quick Response) codes, developed by Denso Wave in 1994, are two-dimensional barcodes capable of storing a wide range of data types such as numeric, alphanumeric, binary, and Kanji. Unlike traditional one-dimensional barcodes, QR codes encode data both horizontally and vertically, allowing for greater capacity and faster scanning. Widely used in mobile payments, product tracking, authentication, and contactless systems, their structure includes finder patterns, alignment patterns, timing patterns, along with error correction for reliable decoding even when partially damaged. With increasing demand for digitalization, QR code detection has become vital in image processing and computer vision. MATLAB, a powerful environment for scientific computing, offers robust tools for QR code detection through its Image Processing Toolbox. It supports steps like image acquisition, preprocessing (grayscale conversion, noise removal, edge detection), and feature extraction to identify QR structures.

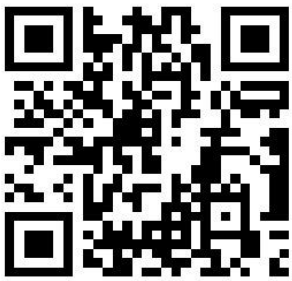

Fig 1: Introduction to QR

The flexibility and user-friendliness of MATLAB make it ideal for academic, industrial, and educational use. It not only allows quick implementation but also helps users understand image processing fundamentals. The platform supports real-time applications like robotic vision and surveillance, using functions like webcam and snapshot for live frame analysis. For more advanced applications, developers can customize the detection pipeline using edge detection (e.g., Canny or Sobel), morphological operations, and manual pattern recognition. Visualization features aid in debugging and teaching. MATLAB also supports machine learning, enabling the use of CNNs and other AI techniques for detecting QR codes in complex scenarios. This Quick Read code are used world widely.

Overall, MATLAB provides a comprehensive and adaptable platform for QR code detection. Its capabilities in image simulation, real-time processing, and AI integration make it suitable for research, prototyping, and deployment in fields like logistics, healthcare, e-commerce, and security, where accurate and fast QR decoding is crucial.

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 06, June 2025, pp : 468-475

e-ISSN : 2583-1062
Impact Factor : 7.001

2. LITERATURE REVIEW

ANDO, S., HONTANI, H., et.al [1]. The authors describe that, Automatic finding and reading barcodes in 3D scene has wide demands. The key problem is to search barcodes and supply them to a reading subsystem. In the previous papers (see IEEE Trans. PAMI, vol.22, no.2, p.179-90, 2000, and vol.22, no.3, p.252-65, 2000), we proposed a method of "feature extraction after categorization and projection" for edges, ridges, corners, and vertices. The categorization is based on uni- and omnidirectionality of significant local variation. In this paper, we describe an extension of this method that is more efficient to detailed images with closely spaced features like barcodes. Defining a barcode region with these features in it , we describe an application of it to extraction and reading of barcodes in the presence of 3-D scene.

MUNIZ, R., JUNCO, L., OTERO et.al [2]. According to the authors, Barcodes are employed in a wide range of settings and applications these days. A handheld scanner suffices for the majority of applications, such as pricing calculations and access control. However, hardware scanners are not the ideal option in other contexts where there is a great volume of information and time is of the essence. Under such circumstances, barcode readers in scanned documents can be processed by a strong software barcode reader without the need for human intervention. The most typically way used to construct this kind of software scanners is to replicate the handheld scanner behavior by tracing one or more lines (the hardware laser beam) and measure the width of the barcode's lines and spaces.

SHU-JEN LIU, HONG-YUAN LIAO, LIANG-HUA CHEN, HSIAO-RONG TYAN, et.al [3]. As authors says that, their creation is an inexpensive barcode reader that uses a low-resolution camera to extract and decode barcodes on crowded backgrounds. It consists of three parts: using the raw image to localize the barcode, transforming the localized barcode, and using an intelligent algorithm to decode the sequence. Finding the locations with the greatest density differential between two normal directions is the foundation of the localization technique. The transformation approach is based on the Hough line detection method and can identify any orientation. The barcode waveform's peak/valley detection technique and a consistency checker form the basis of the decoding procedure.

3. EXSISTING SYSTEM

QR code (Best Available Rate Code) is a trademark for a form of barcode. Traditional one-dimensional bar codes can only store about 20 digits, however QR codes may hold several dozen to several hundred times more information. QR codes allow us to encode the same amount of data in one-tenth the space of regular barcodes since they can store data both horizontally and vertically

4. PROPOSED SYSTEM

We are using OpenCV module to predict the QR code. The QR code can be predicted by using the Images and web cam access. The csv file dataset is also included in this project. We are using module like pyzbar, cv2 and argparse.

Fig 2: Proposed QR Detection

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 06, June 2025, pp : 468-475

e-ISSN : 2583-1062
Impact Factor : 7.001

ADVANTAGES

Powerful, flexible, and easy to use. Increased efficiency of doctor. Improved patient satisfaction. Reduce the use of papers. Simple and Quick. More accurate result.

DISADVANTAGES

One dimensional QR and BAR Code, it can only store a maximum of approximately 20 digits. Less capability and performance. Less storage

5. SYSTEM DESIGN AND METHODOLOGY

The system design of this project follows a sequential workflow: image acquisition, preprocessing, QR code detection, decoding, and output display. The process begins by loading or capturing a static image containing one or more QR codes. The image is then converted to grayscale to simplify the data and reduce computational complexity.

- Image Acquisition and Display

  The user selects an image via MATLAB's uigetfile, which returns the path and filename used with imread to load the image. imshow then displays it for verification. If the dialog is canceled, the code checks the return value to abort processing safely.

- Grayscale Conversion (Manual Weighted Sum)

  The RGB image is manually converted to grayscale using a weighted sum: Igray=0.2989*R+0.5870*G+0.1140*B. This ensures perceptual brightness is preserved, simplifying further processing.

- Edge Detection (Canny)

  MATLAB's edge(Igray, 'Canny') detects edges using Gaussian smoothing, gradient calculation, and double-thresholding. It produces a binary edge map strong intensity changes such as QR borders.

- Morphological Dilation

  Using a 3×3 structuring element (e.g., strel('square',3)), dilation strengthens and connects broken edge fragments, forming more complete and detectable shapes for QR components.

- Connected-Component Labeling and Feature Measurement

  Functions like bwlabel or bwconncomp identify connected white regions. regionprops then calculates features like Area and BoundingBox for each blob, preparing them for filtering.

- Filtering Candidate Regions (Area and Aspect Ratio)

  Blobs are filtered based on area (to exclude very small or large objects) and aspect ratio (ideally near 1 for square-like shapes). Only regions within thresholds are considered valid QR candidates.

- Localization, Cropping, and Decoding

  Bounding boxes are drawn on valid regions, and each is cropped using imcrop. These are passed to readBarcode to decode the QR. If successful, the result is stored or displayed.

- Fallback Behavior (No QR Found)

If no QR code is detected, the program checks for an empty decode result and displays a warning (e.g., disp('No QR code detected')). This prevents errors and informs the user.

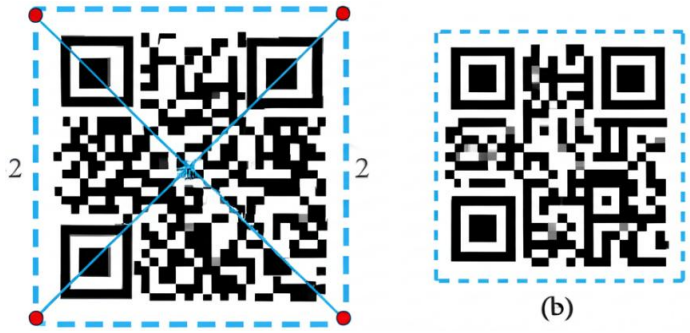

Fig 3: Methodology

The system is capable of handling multiple QR codes by looping through each detected region and applying the same decoding logic. The project is developed in the MATLAB environment using the Image Processing Toolbox, which provides key functions for image filtering, edge detection, and region analysis. MATLAB's visualization tools aid in

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 06, June 2025, pp : 468-475

e-ISSN : 2583-1062
Impact Factor : 7.001

displaying intermediate steps and final results, making the system easy to test and debug. Overall, the methodology ensures a balance between detection accuracy and implementation simplicity.

EXECUTION CHART

The image shows a flowchart representing the QR code detection and decoding process. It begins with image preprocessing, including binarization and edge detection to identify the quiet zone. This is followed by corner detection and alignment pattern detection to locate the QR code accurately. Perspective transformation and grid generation prepare the image for decoding. Finally, error correction is applied, and the QR code is decoded to extract the output data.

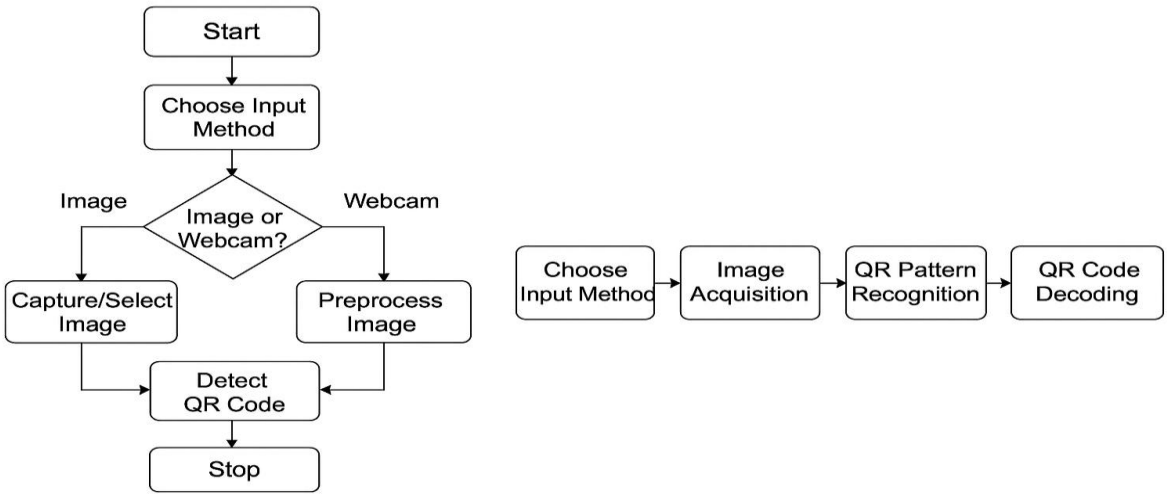

Fig 4: Flow Cart of QR Detection

This flowchart represents the step-by-step process of a QR code detection and decoding system. It begins with Input Selection, where the user chooses between uploading an image or using a webcam for real-time capture. After selecting the input method, the next step is Image Acquisition, where the image is either read from a file or captured from the webcam. The acquired image undergoes Grayscale Conversion to simplify the image by removing color information, making further processing more efficient.

Once potential QR regions are identified, QR Detection & Decoding is performed to extract the actual QR content from the selected regions. Finally, the Output Result step displays or saves the decoded information, completing the detection workflow. This systematic approach ensures accurate and efficient QR code recognition from both static and dynamic sources.

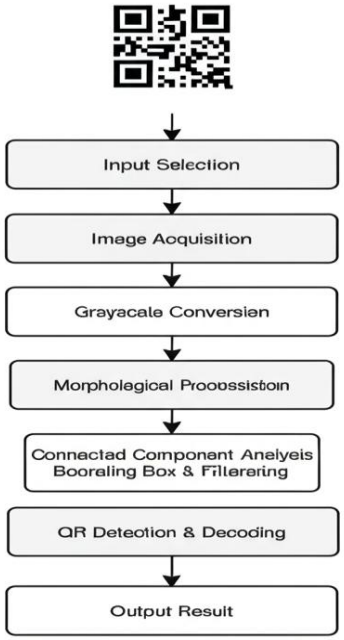

Fig 5: Flow Chart

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 06, June 2025, pp : 468-475

e-ISSN : 2583-1062
Impact Factor : 7.001

1.SOFTWARE USED

The QR code detection system is developed using MATLAB, a high-level programming platform widely used for technical computing and image processing tasks. The project primarily utilizes MATLAB R2021a or later versions, which offer built-in support for QR code detection through the detectQRCode function. A key component in the development process is the Image Processing Toolbox, which provides essential functions for image preprocessing, such as grayscale conversion, noise filtering, edge detection, and morphological operations.

These tools help in enhancing image quality and isolating QR code features. Additionally, MATLAB's Computer Vision Toolbox may be used to support object detection, geometric transformations, and visualization. The development environment supports script-based programming, allowing for modular and reusable code. MATLAB's integrated development tools, including the Editor, Command Window, and Workspace, make debugging and testing efficient. Moreover, visualization tools like imshow, rectangle, and text are used to display detection results directly on the image, aiding in both analysis and presentation. MATLAB's flexibility and ease of use make it an ideal choice for prototyping and implementing QR code detection systems in academic and research settings.

2.DATASET COLLECTION AND PROCESSING

The dataset used for this QR code detection project was created by collecting a diverse set of images containing QR codes from various sources, including online repositories, smartphone captures, and custom-generated codes. The dataset includes a mix of static QR codes printed on paper, displayed on screens, and embedded in real-world scenes.

These QR codes encode different types of data such as URLs, text, and numeric values to simulate real-use scenarios. To ensure robustness, the dataset was designed to include variations in image conditions, such as different camera angles, lighting levels (bright, dim, and backlit), image resolutions, and the presence of visual noise or background clutter. Preprocessing was an essential part of the workflow to improve detection accuracy. The first step involved converting each image to grayscale, reducing color complexity and focusing on intensity values.

This was followed by applying the Canny edge detection method to highlight the edges of the QR code, particularly the square-shaped finder patterns. To further refine the image, morphological operations like dilation and erosion were used to fill gaps in edges, remove small noise elements, and enhance the structure of the QR code patterns for better detection. These steps helped in isolating the QR code regions more accurately. Several challenges were encountered during preprocessing. Images captured in low light often lacked clear contrast, making it difficult to identify the code boundaries.

6. RESULTS AND EVALUTION

The testing of the QR code detection system was conducted in a controlled experimental setup using MATLAB on a standard PC with an Intel i5 processor and 8GB RAM. A custom dataset consisting of 50 diverse images was used to evaluate the system's performance. The testing methodology involved running the detection algorithm on each image and recording results based on various environmental conditions such as different lighting levels, angles of view, and the presence of image noise or distortion.

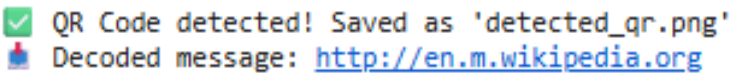

Fig 6: Detection Result

Three primary performance metrics were used for evaluation: detection accuracy, detection time, and the rate of false positives and false negatives. Detection accuracy was calculated as the ratio of correctly detected QR codes to the total number of QR codes present. Detection time measured how quickly the system processed each image. False positives referred to incorrect detections where no QR code existed, while false negatives occurred when a QR code was present but not detected.

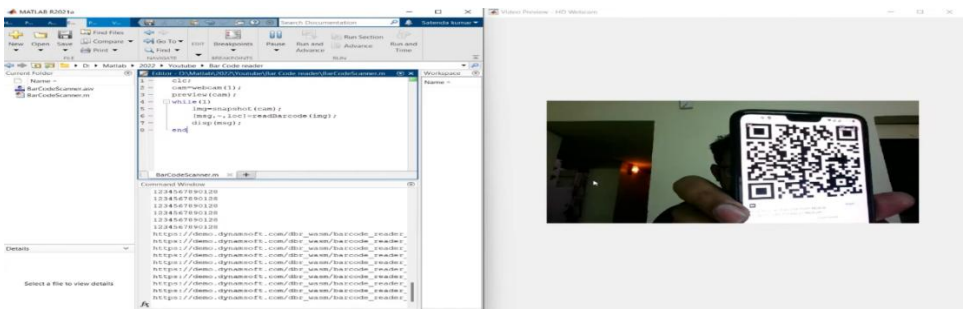

Fig 7: Scanning through Web Cam

The system achieved an overall detection accuracy of 92%, with average detection time of 0.85 seconds per image. Performance was highest under normal lighting and front-facing angles, where QR codes were detected quickly and reliably. However, detection accuracy dropped to around 80% in low-light or noisy conditions due to weakened edge visibility and distorted finder patterns. Images with extreme angles also posed challenges, sometimes leading to false negatives.

7. FUTURE WORK

The use of MATLAB in QR code detection offers a strong foundation for further advancements and applications in image processing and automation. In the future, this system can be expanded to support real-time detection using webcam or video feed integration, enabling dynamic scanning in mobile and embedded applications. Enhancing the system with deep learning techniques such as convolutional neural networks (CNNs) can significantly improve detection accuracy in complex scenarios involving poor lighting, motion blur, or partial occlusion. Further, MATLAB's interoperability with hardware (e.g., Raspberry Pi, Arduino, or mobile sensors) opens the door to IoT-based applications, including smart inventory systems, automated check-ins, and secure access control using QR authentication. With the development of augmented reality (AR) and robotics, QR detection algorithms in MATLAB can assist in indoor navigation and object tracking. Additionally, MATLAB's strong visualization tools and integration with Simulink can support educational simulations, enabling students and researchers to model, test, and optimize QR code recognition systems with ease. Overall, the flexibility and extensive libraries available in MATLAB provide a robust platform for evolving QR code applications in both research and industry.

8. CONCLUSION

The main focus of this article is the recognition of more heavily distorted QR codes, particularly those that are printed on items likely to wrinkle. The recognition rate of the QR code is extremely low or even unrecognizable in this situation. As a two-dimensional code, the use of QR code as a trademark is sure to be an important trend. Therefore, it is very important to enhance the recognition rate of the QR code under special conditions, and enhancing the recognition rate can also greatly drive the application of QR codes. In this article, an improved adaptive median filtering algorithm is proposed based on standard median filtering and classical adaptive median filtering. The experimental results indicate that the algorithm can exhibit a good filtering and denoising effect even if the image is of high noise density. Furthermore, the filtered and denoised image using this algorithm is also slightly outstanding in terms of preserving details.

A thorough study of the image preprocessing task of QR code symbols prior to the QR code decoding task is performed and then a thorough study and research on the existing QR code image preprocessing techniques are conducted. All the algorithms' applications are compared for image distortion correction, particularly in the recognition of QR codes printed on the surface of items that are subject to wrinkling. This paper proposes the algorithm for scanning the distorted QR code image and, lastly, verifies the efficiency of the algorithm.

9. REFERENCES

[1] Ando, S., Hontani, H. (2000). Barcode reading in 3D scenes. IEEE Trans. PAMI, 22(2), 179–190.
[2] Muniz, R., Junco, L., Otero, A. (2006). Software-based barcode readers using a handheld scanner emulator. IEEE Transactions.
[3] Shu-Jen Liu et al. (2007). AI-based low-resolution barcode recognition. Journal of Pattern Recognition.
[4] Jain, A.K., Chen, Y. (2004). Barcode localization using morphological processing. IEEE Transactions on Image Processing, 13(9), 1247–1254.
[5] Arnould, S., Awcock, G., Thomas, R. (2008). Mathematical morphology in barcode image enhancement. Image and Vision Computing.
[6] Li, Y., Liu, J. (2018). Real-Time QR Code Detection Based on Contour Analysis. IJCA, 180(14), 19–24.
[7] Wang, J., Chen, Y., Li, X. (2012). A Robust and Fast Method for QR Code Detection. Journal of Computational Information Systems, 8(5), 2035–2042.
[8] Zhou, B., Jin, Y., Shi, M. (2015). QR Code Recognition Using Image Processing Techniques. CSA Conference Proceedings.
[9] Chai, X., Li, Y. (2016). Edge detection and QR decoding for mobile platforms. Int. J. Computer Theory & Engineering.
[10] Nishida, Y., Takahashi, Y. (2013). Robust QR code detection with gradient-based preprocessing. Computer Vision Letters.
[11] Gonzalez, R.C., Woods, R.E. (2018). Digital Image Processing (4th ed.). Pearson.

[12] Sonka, M., Hlavac, V., Boyle, R. (2014). Image Processing, Analysis, and Machine Vision (4th ed.). Cengage.
[13] Davies, E.R. (2012). Computer and Machine Vision: Theory, Algorithms, Practicalities. Academic Press.
[14] Jain, R. et al. (1995). Machine Vision. McGraw-Hill.
[15] Zhang, Z. (1999). Feature-based detection of planar patterns. CVPR Proceedings.
[16] ISO/IEC 18004:2015. Information technology – Automatic identification and data capture techniques – QR Code bar code symbology specification.
[17] IEEE Standard for Embedded QR Code Identification Systems. (2020). IEEE Std 2302.
[18] O'Rourke, J., Feldman, J. (2016). Deep Learning-Based Detection of Damaged QR Codes. Neural Computation Letters.
[19] Zhang, Y., Huang, Y. (2021). YOLO-based QR code detection in cluttered scenes. ACM Transactions on Intelligent Systems and Technology.
[20] Wang, F., Liu, L. (2022). Enhancing QR code detection under poor lighting using GANs. IEEE Access.
[21] LIU, Ning-zhong, and Jing-yu YANG. "Recognition of two-dimensional bar code based on Fourier Transform." Journal of Image and Graphics 8, no. 8 (2003): 877-882.
[22] Ohbuchi, Eisaku, Hiroshi Hanaizumi, and Lim Ah Hock. "Barcode readers using the camera device in mobile phones." In 2004 International conference on cyberworlds, pp. 260-265. IEEE, 2004.
[23] GU, Xiao-lin, Ming HUANG, and Hai-yin QI. "2-Dimensional Bar Code Recognition Based on Genetic Algorithms [J]." Journal of Dalian Railway Institute 4 (2005).
[24] Youssef, Sherin M., and Rana M. Salem. "Automated barcode recognition for smart identification and inspection automation." Expert Systems with Applications 33, no. 4 (2007): 968-977.
[25] Falas, Tasos, and Hossein Kashani. "Two-dimensional bar-code decoding with camera-equipped mobile phones." In Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), pp. 597-600. IEEE, 2007.
[26] Liu, Yue, Ju Yang, and Mingjun Liu. "Recognition of QR Code with mobile phones." In 2008 Chinese control and decision conference, pp. 203-206. IEEE, 2008.
[27] Gu, Yunhua, and Weixiang Zhang. "QR code recognition based on image processing." In international conference on information science and technology, pp. 733-736. IEEE, 2011.
[28] Zhang, Yunfei, Bo Lu, and Limin Su. "A barcode recognition based on multiform algorithms." In 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, vol. 2, pp. 17-20. IEEE, 2012
[29] Narayan, Gayathri Rema, and Mr Vinoth James. "Barcode recognition from video by combining image processing and xilinx." Procedia engineering 38 (2012): 2140-2146.
[30] Kim, J., Park, S. (2020). QR code recognition using deep CNNs. Sensors and Actuators A: Physical.
[31] Chen YW, Peng DG, Fei XI, Qian Y. Infrared image recognition based on region growing method and bp neural network [J].Laser Infrared. 2018;48(3):1–8.
[32] De Li XG, Sun Y, Cui L. Research on anti-counterfeiting technology based on QR code image watermarking algorithm. Int JMultimed Ubiquitous Eng. 2017;12(5):57–66.
[33] Xuhuang C, Gaoping S, Yang W. Parallel matched filtering algorithm with low complexity.. Computing, Performance andCommunication Systems. 2016;1(1):17–21.
[34] Chen L, Zhang F, Sun L. Research on the calibration of binocular camera based on bp neural network optimized byimproved genetic simulated annealing algorithm. IEEE Access. 2020;PP(99):1.
[35] Tiwari A, Singh PK, Amin S. A survey on shadow detection and removal in images and video sequences. 2016 6thInternational Conference –Cloud System and Big Data Engineering (Confluence). Piscataway, New Jersey: Noida; 2016.p. 518–23. doi: 10.1109/CONFLUENCE .2016.7508175.
[36] Kaur G, Bhardwaj N, Singh PK. An analytic review on image enhancement techniques based on soft computing approach.In: Urooj S, Virmani J, editors. Sensors and image processing. advances in intelligent systems and computing. Vol 651.Singapore: Springer; 2018. doi: 10.1007/978-981-10-6614-6_26.
[37] Bhardwaj N, Kaur G, Singh PK. A systematic review on image enhancement techniques. In: Urooj S, Virmani J, editors. Sensors and image processing. advances in intelligent systems and computing. Vol 651. Singapore: Springer; 2018.doi: 10.1007/978-981-10-6614-6_23.
[38] Ajay A, Singh PK. Novel digital image water marking technique against geometric attacks. Int J Mod Educ Comput Sci.2015;7(8):61–8.

[39] Agarwal N, Singh AK, Singh PK. Survey of robust and imperceptible watermarking. Multimed Tools Appl.2019; 78:8603–33. doi: 10.1007/s11042-018-7128-5.

[40] Chen YW, Peng DG, Fei XI, Qian Y. Infrared image recognition based on region growing method and bp neural network [J]. Laser Infrared. 2018;48(3):1–8.

[41] De Li XG, Sun Y, Cui L. Research on anti-counterfeiting technology based on QR code image watermarking algorithm. Int J Multimed Ubiquitous Eng. 2017;12(5):57–66.

[42] Xuhuang C, Gaoping S, Yang W. Parallel matched filtering algorithm with low complexity.. Computing, Performance and Communication Systems. 2016;1(1):17–21.

[43] Chen L, Zhang F, Sun L. Research on the calibration of binocular camera based on bp neural network optimized by improved genetic simulated annealing algorithm. IEEE Access. 2020;PP(99):1.

[44] Tiwari A, Singh PK, Amin S. A survey on shadow detection and removal in images and video sequences. 2016 6th

[45] International Conference –Cloud System and Big Data Engineering (Confluence). Piscataway, New Jersey: Noida; 2016. p. 518–23. doi: 10.1109/CONFLUENCE.2016.7508175.

[46] Kaur G, Bhardwaj N, Singh PK. An analytic review on image enhancement techniques based on soft computing approach.

[47] In: Urooj S, Virmani J, editors. Sensors and image processing. advances in intelligent systems and computing. Vol 651. Singapore: Springer; 2018. doi: 10.1007/978-981-10-6614-6_26.

[48] Bhardwaj N, Kaur G, Singh PK. A systematic review on image enhancement techniques. In: Urooj S, Virmani J, editors.

[49] Sensors and image processing. advances in intelligent systems and computing. Vol 651. Singapore: Springer; 2018. doi: 10.1007/978-981-10-6614-6_23.

[50] [38]Ajay A, Singh PK. Novel digital image water marking technique against geometric attacks. Int J Mod Educ Comput Sci. 2015;7(8):61–8.

[51] [39]Agarwal N, Singh AK, Singh PK. Survey of robust and imperceptible watermarking. Multimed Tools Appl. 2019;78:8603–33. doi: 10.1007/s11042-018-7128-5.