Dear Author,

Here are the proofs of your article.

- You can submit your corrections **online**, via **e-mail** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- You can also insert your corrections in the proof PDF and **email** the annotated PDF.
- For fax submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the **journal title**, **article number**, and **your name** when sending your response via e-mail or fax.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/ corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript.*
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style.
  Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.
- Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**
- The **printed version** will follow in a forthcoming issue.

Please note

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL: http://dx.doi.org/[DOI].
If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information go to: http://www.link.springer.com.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us if you would like to have these documents returned.

Metadata of the article that will be visualized in OnlineFirst

ArticleTitle	A comprehensive study on enhanced QR extraction techniques with deep learning-based verification	
Article Sub-Title		
Article CopyRight	The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature (This will be the copyright line in the final PDF)	
Journal Name	Applied Intelligence	

Corresponding Author	FamilyName	**Moon**
	Particle	
	Given Name	**Hyeonjoon**
	Suffix	
	Division	Department of Computer Science and Engineering
	Organization	Sejong University
	Address	Seoul, 05006, Republic of Korea
	Phone	
	Fax	
	Email	hmoon@sejong.ac.kr
	URL	
	ORCID	
Author	FamilyName	**Alam**
	Particle	
	Given Name	**Nur**
	Suffix	
	Division	Department of Computer Science and Engineering
	Organization	Sejong University
	Address	Seoul, 05006, Republic of Korea
	Phone	
	Fax	
	Email	nur0756@sju.ac.kr
	URL	
	ORCID	
Author	FamilyName	**Sagar**
	Particle	
	Given Name	**A S M Sharifuzzaman**
	Suffix	
	Division	Department of Intelligent Mechatronics Engineering
	Organization	Sejong University
	Address	Seoul, 05006, Republic of Korea
	Phone	
	Fax	
	Email	sharifsagar80@sejong.ac.kr
	URL	
	ORCID	
Author	FamilyName	**Zhang**
	Particle	
	Given Name	**Wenqi**
	Suffix	
	Division	Department of Computer Science and Engineering

	Organization	Sejong University
	Address	Seoul, 05006, Republic of Korea
	Phone	
	Fax	
	Email	
	URL	
	ORCID	
Author	FamilyName	**Jin**
	Particle	
	Given Name	**Taicheng**
	Suffix	
	Division	Department of Computer Science and Engineering
	Organization	Sejong University
	Address	Seoul, 05006, Republic of Korea
	Phone	
	Fax	
	Email	
	URL	
	ORCID	
Author	FamilyName	**Dosset**
	Particle	
	Given Name	**Arailym**
	Suffix	
	Division	Department of Computer Science and Engineering
	Organization	Sejong University
	Address	Seoul, 05006, Republic of Korea
	Phone	
	Fax	
	Email	
	URL	
	ORCID	
Author	FamilyName	**Dang**
	Particle	
	Given Name	**L. Minh**
	Suffix	
	Division	Institute of Research and Development
	Organization	Duy Tan University
	Address	Da Nang, 550000, Viet Nam
	Division	Faculty of Information Technology
	Organization	Duy Tan University
	Address	Da Nang, 550000, Viet Nam
	Phone	
	Fax	
	Email	
	URL	
	ORCID	

Schedule	Received	
	Revised	
	Accepted	23 Mar 2025

Abstract

In the digital age, Quick Response (QR) codes have become essential in sectors such as digital payments and ticketing, propelled by advancements in Internet of Things (IoT) and deep learning. Despite their growing use, there are significant challenges in the accurate extraction and verification of QR codes, particularly in dynamic environments. Traditional methods struggle with issues like variable lighting, complex backgrounds, and counterfeits, which degrade the performance of QR code extraction and verification processes. This paper introduces a comprehensive approach that refines QR code extraction using enhanced adaptive thresholding techniques and incorporates a deep learning framework specifically tailored for robust QR code verification. Our methodology integrates dynamic window size adjustment, statistical weighting, and post-thresholding refinement to ensure precise QR code extraction under varying conditions. The verification process employs the ShuffleNetV2 network to ensure high performance with significantly low processing times suitable for real-time applications. Additionally, our deep learning model is trained on a comprehensive dataset comprising 28,523 images across 24 distinct QR code pattern classes, including variations in lighting, noise, and backgrounds to simulate real-world conditions. Experimental results demonstrate that our proposed methodology outperforms competing techniques in both processing speed and recognition accuracy, achieving a processing time of 0.08 seconds and a validation accuracy of 99.99% in constrained scenarios. Our approach shows an exceptional ability to distinguish real QR codes from counterfeits and highlights the significance and efficacy of our method in addressing contemporary challenges.

Keywords (separated by '-')	QR code extraction - Deep learning - Feature extraction - Verification
Footnote Information	

Applied Intelligence _####################_
https://doi.org/10.1007/s10489-025-06509-y

A comprehensive study on enhanced QR extraction techniques with deep learning-based verification

Nur Alam[1] · A S M Sharifuzzaman Sagar[2] · Wenqi Zhang[1] · Taicheng Jin[1] · Arailym Dosset[1] · L. Minh Dang[3,4] · Hyeonjoon Moon[1]

Accepted: 23 March 2025

Abstract

In the digital age, Quick Response (QR) codes have become essential in sectors such as digital payments and ticketing, propelled by advancements in Internet of Things (IoT) and deep learning. Despite their growing use, there are significant challenges in the accurate extraction and verification of QR codes, particularly in dynamic environments. Traditional methods struggle with issues like variable lighting, complex backgrounds, and counterfeits, which degrade the performance of QR code extraction and verification processes. This paper introduces a comprehensive approach that refines QR code extraction using enhanced adaptive thresholding techniques and incorporates a deep learning framework specifically tailored for robust QR code verification. Our methodology integrates dynamic window size adjustment, statistical weighting, and post-thresholding refinement to ensure precise QR code extraction under varying conditions. The verification process employs the ShuffleNetV2 network to ensure high performance with significantly low processing times suitable for real-time applications. Additionally, our deep learning model is trained on a comprehensive dataset comprising 28,523 images across 24 distinct QR code pattern classes, including variations in lighting, noise, and backgrounds to simulate real-world conditions. Experimental results demonstrate that our proposed methodology outperforms competing techniques in both processing speed and recognition accuracy, achieving a processing time of 0.08 seconds and a validation accuracy of 99.99% in constrained scenarios. Our approach shows an exceptional ability to distinguish real QR codes from counterfeits and highlights the significance and efficacy of our method in addressing contemporary challenges.

Keywords QR code extraction · Deep learning · Feature extraction · Verification

✉ Hyeonjoon Moon
hmoon@sejong.ac.kr

Nur Alam
nur0756@sju.ac.kr

A S M Sharifuzzaman Sagar
sharifsagar80@sejong.ac.kr

1 Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea

2 Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Republic of Korea

3 Institute of Research and Development, Duy Tan University, Da Nang 550000, Viet Nam

4 Faculty of Information Technology, Duy Tan University, Da Nang 550000, Viet Nam

1 Introduction

Quick Response (QR) codes, two-dimensional barcodes, have become an indispensable component of the contemporary digital ecosystem. Their ability to store substantial data and offer rapid scanning makes them pivotal in sectors ranging from payments to ticketing and marketing. As Industrial Internet of Things (IoT) and deep learning technologies advance [1–3], QR codes serve as cost-effective reading labels, especially in high-demand settings such as COVID-19 testing centers and logistics hubs. However, despite their widespread utility, they are not without challenges. Motion blur, uneven lighting, and issues in dynamic environments, particularly where mobile robots operate, underscore the complexities of QR code recognition in our technologically advanced age (Figs. 1).

Journal: 10489 MS: 6509 TYPESET DISK LE CP Disp.:2025/3/28 Pages: 22 Layout: Large

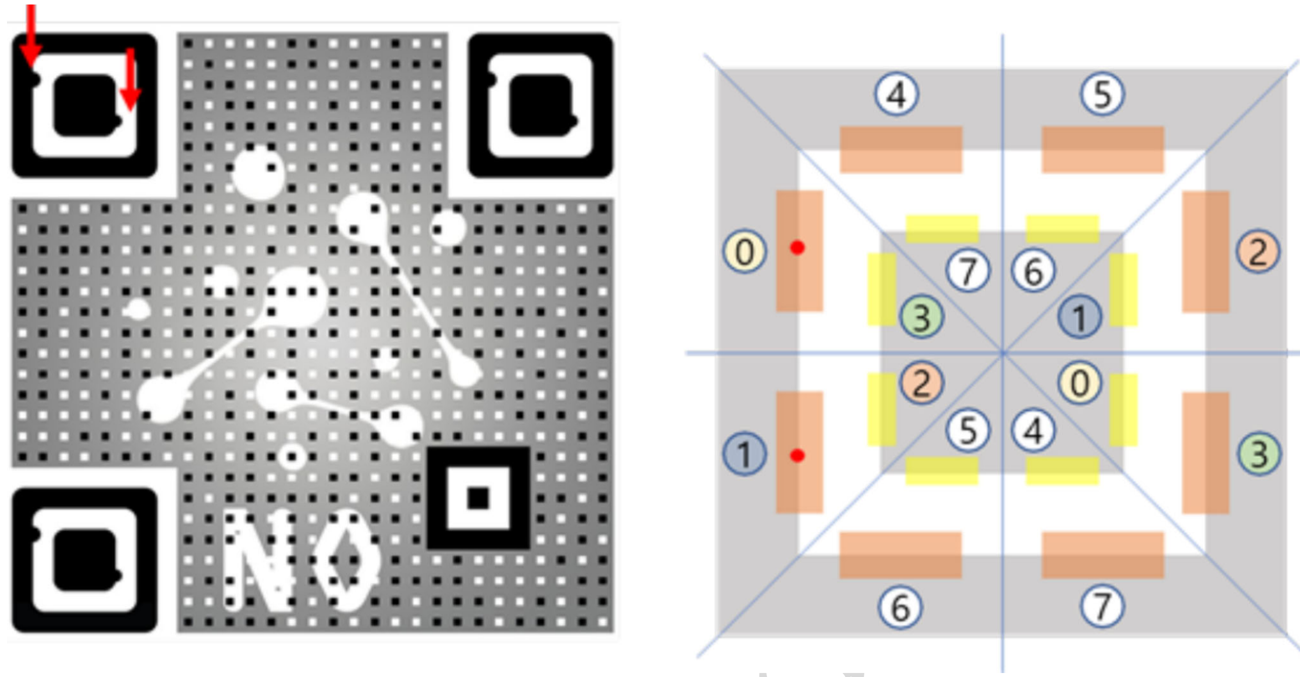


Fig. 1 Sample of an authentic QR code illustrating its intricate pattern design components. Each segment represents the structural elements integral to its uniqueness and readability

Historically, the journey of QR code recognition has been marked by continuous evolution. Initial recognition methods leaned heavily on image-processing techniques, which, while groundbreaking in their time, faced significant challenges. Uneven illumination, highlight spots, and complex backgrounds often degraded QR code readability. Techniques like Otsu's thresholding were effective for images with simple backgrounds but faltered under varying conditions [4]. Blanger and Hirata enhanced QR code recognition in natural scenes using a modified Single Shot Detector that incorporates subpart annotations [5]. While effective for individual QR code identification, their approach was less suited for batch processing in dense environments. Jiang et al. addressed this limitation with their app, which specifically improves handling densely arranged QR codes through an adaptive code detection mechanism and a novel image refocus technique but struggled with code detection in extremely small or closely spaced scenarios [6]. He and Yang improved upon previous methods by implementing an adaptive binarization method that dynamically adjusts to lighting conditions, enhancing QR code image processing under uneven illumination [7]. However, their method's complexity increases computational demands due to the necessity for adaptive window sizing and threshold calculations. Zhang et al. further advanced this field by developing a region-based network capable of finely localizing and classifying multi-class barcodes in complex environments [8]. Their approach, which integrates multi-scale spatial pyramid pooling and quadrilateral bounding box regression, effectively handles small-scale barcodes and distortions but introduces complexity in terms of computational overhead. Dong et al. improved previous works by introducing a generative adversarial network combined with an attention mechanism to recognize motion-blurred QR codes, significantly improving processing time and recognition accuracy [9]. As the field progressed, there was a shift towards more advanced strategies, such as morphological processing, which, despite being computationally intensive, aimed to tackle more intricate backgrounds. However, many of these methods had a narrow focus, often limited to specific QR code scenarios, which proved inadequate in diverse environments.

Apart from these, QR code verification is also an important field after recognizing QR code patterns. Recent studies mainly use AI-based approaches such as convolutional neural networks [10–12] for QR code verification. Yan et al. introduced an IoT-based anti-counterfeiting system that integrates visual features with QR codes to enhance security by utilizing natural and printed micro-features for robust verification [13]. Ismail et al. developed a QR code validation method to improve QR code security by integrating advanced URL analysis to block malicious and phishing URLs effectively [14]. Their method adds robust phishing detection rules and leverages multiple validation layers to safeguard against sophisticated cyber threats, albeit at the expense of increased complexity in validation processes. Cu Vinh Loc et al. introduced a QR code verification method

Journal: 10489 MS: 6509 TYPESET DISK LE CP Disp.:2025/3/28 Pages: 22 Layout: Large

using digital watermarking and a Siamese neural network to ensure authenticity, achieving high accuracy but at the cost of increased computational complexity [15]. Loc et al. further developed a tamper-proof QR code system using a deep learning-based data hiding method that embeds a secret security feature within the QR code, verified through a deep neural network and Siamese network analysis [16]. This approach enhances security against QR code tampering and offers high accuracy but requires significant computational resources for its dual-network architecture. Hantono et al. presented a novel system for counterfeit detection using multi-featured secure 2D grayscale codes [17]. This approach significantly enhances counterfeit detection by incorporating spatial and frequency domain analyses and grayscale watermarking to assess image quality degradation. Despite its high precision and specificity, the complexity of its multi-feature analysis could present scalability and computational challenges in real-world applications. Moreover, these methods only consider very limited patterns for the verification process.

In this paper, we address long-standing challenges in the domains of QR code extraction and verification with a comprehensive and innovative approach. Building on the foundation of traditional methods, our methodology enhances adaptive thresholding techniques, introducing refinement algorithms that effectively counter common image disturbances such as noise and uneven illumination. Our approach goes beyond extraction; we have integrated state-of-the-art edge detection and contour extraction algorithms tailored for discerning intricate QR code patterns, even in cluttered environments. Furthermore, we employ a deep learning framework meticulously trained on large datasets. This ensures not only structural validation of QR codes but also a deeper examination of their authenticity, setting our approach apart in ensuring data integrity and security.

2 Related works

2.1 QR code extraction

QR codes, initially designed for tracking automotive parts, have expanded to various applications, from mobile payments to augmented reality. This diversification has increased the demand for advanced extraction techniques [18]. Traditional extraction methods relied heavily on image processing strategies such as thresholding, morphological operations, and edge-based contour detection. However, these methods often faltered in diverse imaging scenarios, especially with challenges such as variable lighting, complex backgrounds, and varying orientations.

Several methodologies have been introduced to address these limitations. Ohbuchi et al. utilized the intrinsic Digital Signal Processor (DSP) of the QR code for location discernment [19]. Although effective in certain scenarios, this method struggles with QR codes that have damaged or obscured DSPs. Hu et al. differentiated texture differences between QR codes and backgrounds [20]. The performance of their proposed method is degraded by complex or noisy backgrounds. Dubská et al. [21] and Gabriel [22] used the Hough transform and parallel line detection, respectively. These methods, while innovative, were susceptible to errors in images with multiple parallel or perpendicular lines not related to QR codes. The methods in [23] and those of Tingting Huang [24] relied on dilation, erosion, and morphological operators. However, they often had limited detection rates, especially in cluttered environments. Tzu-Han Chou et al. [25] used convolutional neural networks, showcasing the potential of deep learning. However, these methods required substantial computational resources and extensive training data. The method by Hou et al. [26] was optimized for simple image data but could struggle with more complex or degraded QR codes. Ostkamp et al. [27], M. Ahn et al. [28], Y. Kato et al. [29], Liu Y. [30], CH Chu [31], and Qichao Chen [32] focused on improving image quality. While these methods improved readability, they did not always guarantee accurate extraction. The method by Luiz Belussi and Nina S. T. Hirata [33] achieved a commendable detection rate but could not be universally effective in all scenarios.

These gaps in existing methodologies highlight the need for a comprehensive and adaptive extraction strategy, which led to our proposed method. Our approach aims to integrate the strengths of previous techniques while addressing their limitations, offering a balanced solution for QR code extraction.

2.2 QR code verification

The widespread adoption of QR codes in areas such as digital payments and personal data sharing underscored a pressing challenge: the need for robust verification of the authenticity of QR codes. Initial verification strategies, which focused primarily on basic structural checks of QR codes, quickly became obsolete as forgery techniques evolved, leaving a significant gap in the security landscape.

Xie and Tan [34] developed an anti-counterfeiting system that emphasized QR code copy detection. While their approach enhanced the estimation of QR pattern locations in images, it primarily addressed product counterfeits and was not effective for more sophisticated forgeries. The method in [35] utilized the decentralized nature of blockchain combined with smart contracts. Although promising, the complexity and scalability of blockchain solutions can sometimes be a limitation, especially in real-time verification scenarios. Tran and Hong [36] leveraged RFID techniques, focusing on tag authentication. However, the dependency of RFID on specialized hardware can be a constraint. Sim-

Journal: 10489 MS: 6509 TYPESET DISK LE CP Disp.:2025/3/28 Pages: 22 Layout: Large

ilarly, the holography method [37], although innovative, requires specialized equipment and might not be feasible for all applications. Yiu's approach [38], rooted in Near-Field Communications (NFC), provided product origin tracking. Although NFC offers a layer of security, its range limitation and hardware dependency can be restrictive in various scenarios. Krishna and Dugar [39] encrypted the information within QR codes, offering server-side verification. Their method, however, authenticated a QR code only once, which might not be suitable for all use cases. Similarly, Wan et al. [40] combined visual secret sharing with QR codes. Although innovative, reliance on secret visual data might pose challenges in environments with variable lighting or image quality.

In summary, while each of these methods brought unique strengths to the table, they also had inherent limitations. These gaps in existing verification methodologies emphasize the need for a more comprehensive, adaptable, and universally applicable solution, paving the way for our proposed method.

3 Dataset preparation and augmentation

3.1 Dataset preparation

Every identified pattern is given its own processing strategy to guarantee reliable recognition. All three finding patterns are applied using the same round pattern with similar R values at each corner, although the specific R value may change depending on the effectiveness of the product and the solution. To distinguish between genuine and counterfeit products among the 16 patterns considered, this round processing pattern information is stored in a database for comparison with artificial intelligence discriminating patterns. Specifically, to efficiently speed up the authentication process, the first two of these patterns are removed from the 6 dataset (Fig 2).

The data were stored in PNG format with a resolution of 744 × 744 pixels and were collected from Nexpot Solution (https://taglab9.co.kr/). To maintain a constant recognition rate across all three patterns, an identical symbol is added to the finding patterns. The allotted area for this symbol addition may be decreased depending on performance factors. The symbol itself has a semicircular form that occasionally resembles an oval or a trapezoid, depending on the shooting conditions. The increased data storage of symbol patterns in the database facilitates the comparison of discriminating artificial intelligence patterns, streamlining the process of identifying genuine products from counterfeits.

To reduce false recognition rates, selected symbol patterns are used for the parameters. Recognizing that it is challenging to generate every possible fake pattern, strategies are investigated to improve counterfeit pattern identification. The interaction between design and recognition is taken into consideration when determining the ideal symbol size value. Eight different symbol patterns are used in total, with an emphasis on evaluating their performance in terms of recognition rates to ensure that the dataset is useful for QR code authentication (Fig. 3).

Finally, there were a total of 24 different classifications in the dataset. Twenty-three of these classes, each having a unique combination of two patterns and positive values, represented genuine patterns. Additionally, there was a class specifically for counterfeit patterns that served as a guide to distinguish fake patterns from genuine ones, as shown in 7 Table 1.

Data augmentation was done following the instructions in Section 3.2. Using data augmentation techniques, our dataset expanded to include a total of 28,523 pattern images. To

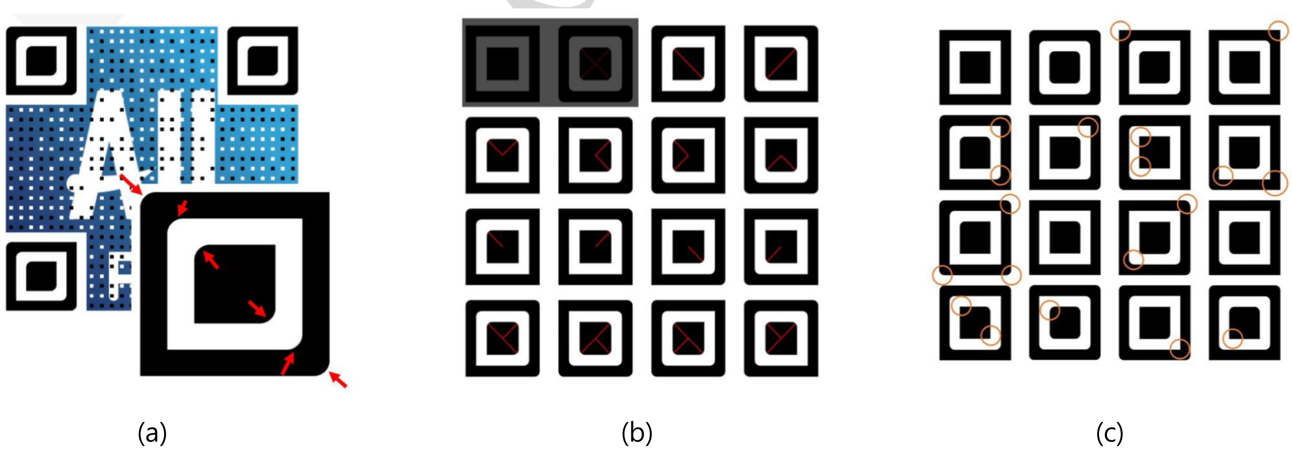


Fig. 2 Illustration of the 16 distinct QR code patterns leveraged for advanced verification, highlighting the unique characteristics of each pattern for robust authentication

Journal: 10489 MS: 6509 TYPESET DISK LE CP Disp.:2025/3/28 Pages: 22 Layout: Large

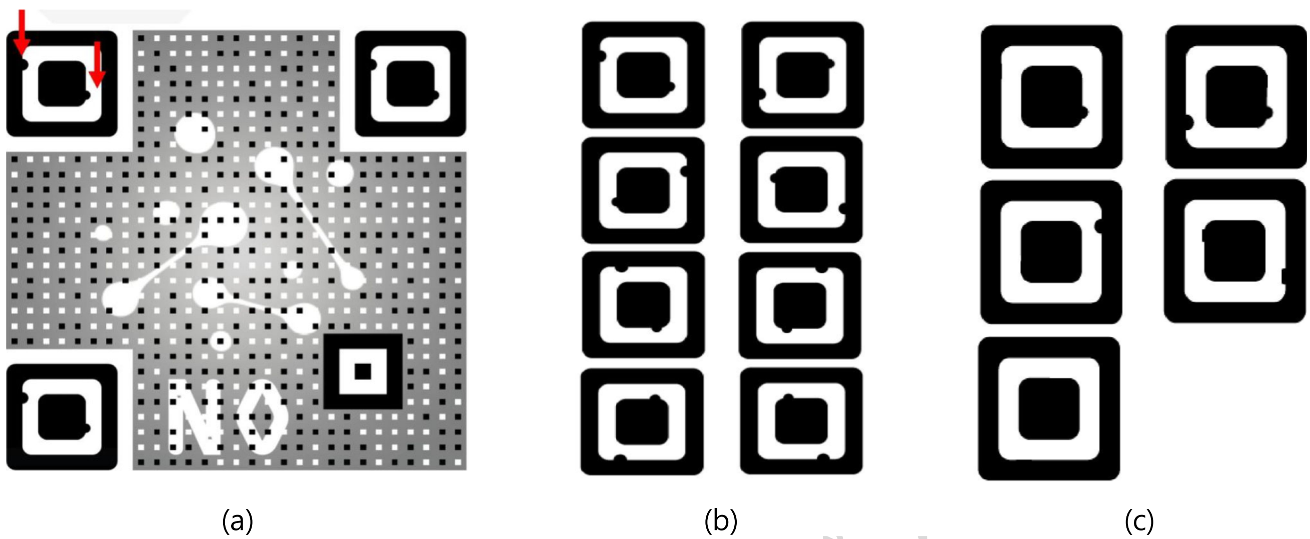


Fig. 3 QR code images of different patterns which were used to divide images into 24 distinct classes based on inherent pattern variations, establishing the foundation for classification and analysis

ensure reliable model training and evaluation, we divided this dataset into two subsets: 80% for training and 20% for validation.

3.2 Data augmentation

Data augmentation refers to methods used to increase the size and introduce diversity into a dataset by making various changes. For classification and verification tasks, data augmentation plays an essential role in training deep learning models and helps prevent overfitting. In the context of our classification model, “data augmentation” includes a variety of modifications, such as cropping, clipping, flipping, perspective adjustments, rescaling, color adjustments, brightness variations, adding occlusions, adding darkness, and rotation. To increase the diversity of the dataset for our study, we specifically implemented color adjustments, brightness variations, occlusions, and darkness adjustments.

The data augmentation methods used on our QR pattern datasets are shown in Fig. 4. These methods not only expand the dataset but also strengthen its resistance to overfitting. We have selected four essential enhancement strategies from among these approaches: color modifications, brightness variations, occlusions, and darkness adjustments. As part of the augmentation procedure, flipping and perspective changes are also performed at random. The resultant augmented images are then utilized to train our QR pattern classification model, enabling it to effectively classify various types of patterns.

Table 1 Distribution of images across 24 classes, detailing the count of images associated with each specific QR code pattern, from FAKE to RI15

Classes	Number of images	Classes	Number of images
FAKE	1030	RI04	1143
PI1	1224	RI05	1092
PI2	1242	RI06	1167
PI3	1182	RI07	1227
PI4	1221	RI08	1209
PI5	1209	RI09	1200
PI6	1185	RI10	1224
PI7	1239	RI11	1140
PI8	1227	RI12	1182
RI01	1173	RI13	1239
RI02	1188	RI14	1212
RI03	1140	RI15	1218

4 Materials and methods

In this section, we introduce the details of our proposed framework for QR code authentication. The overall structure mainly consists of two parts: pattern extraction and QR code authentication using pattern verification techniques. In the pattern extraction part, we employ enhanced adaptive thresholding methods. These techniques collectively improve the QR code extraction process from images with varying conditions, such as complex backgrounds, noise, and variable lighting. In the authentication part, we utilize various data augmentation, feature extraction, and pattern verification techniques to authenticate the QR code.

Journal: **10489** MS: **6509** TYPESET DISK LE CP Disp.:**2025/3/28** Pages: **22** Layout: **Large**

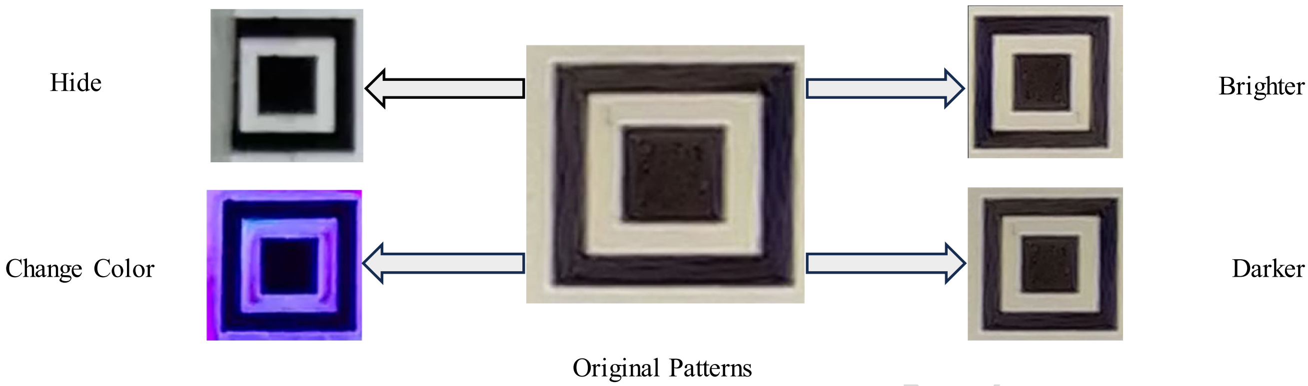


Fig. 4 Visualization of various image augmentation techniques applied to enhance the quality of QR code pattern images in this study

4.1 QR code extraction

Quick Response (QR) codes have surpassed their original application of tracking vehicle components to become widespread in our digital lives. Given their widespread adoption, the need for precise, swift, and adaptable QR code extraction has escalated exponentially. QR code extraction poses several challenges that conventional image processing techniques struggle to overcome. Some of these challenges include variability in scale and orientation, complex backgrounds, and inconsistent lighting (Fig. 5).

Recently, thresholding, an image processing technique, has been used to address some of these challenges. However, the traditional thresholding approach encounters difficulties in scenarios with variable lighting conditions, creating the need for a more fine-tuned method. Adaptive thresholding addresses this by dividing the image into smaller sections and dynamically computing the threshold for each section based on localized characteristics, such as the mean intensity of neighboring pixels. While adaptive thresholding improves upon the rudimentary nature of global thresholding, its performance is also affected by some limitations, especially when used for QR code extraction:

1. **Fixed Window Size:** Traditional adaptive thresholding uses a fixed window size to analyze the local neighborhood, which is ineffective in capturing QR codes of different sizes.
2. **Mean-Only Computation:** Using only the mean intensity value for thresholding can be too simplistic when QR codes are embedded in images with complex patterns or noise.
3. **Lack of Post-Processing:** After the thresholding process, the output often contains artifacts or noise that can hinder subsequent steps like edge detection and contour extraction.

These limitations require modifications to traditional adaptive thresholding algorithms. Our study aims to address these shortcomings by offering a fine-tuned version of adaptive thresholding that is specially optimized for the intricate demands of QR code extraction.

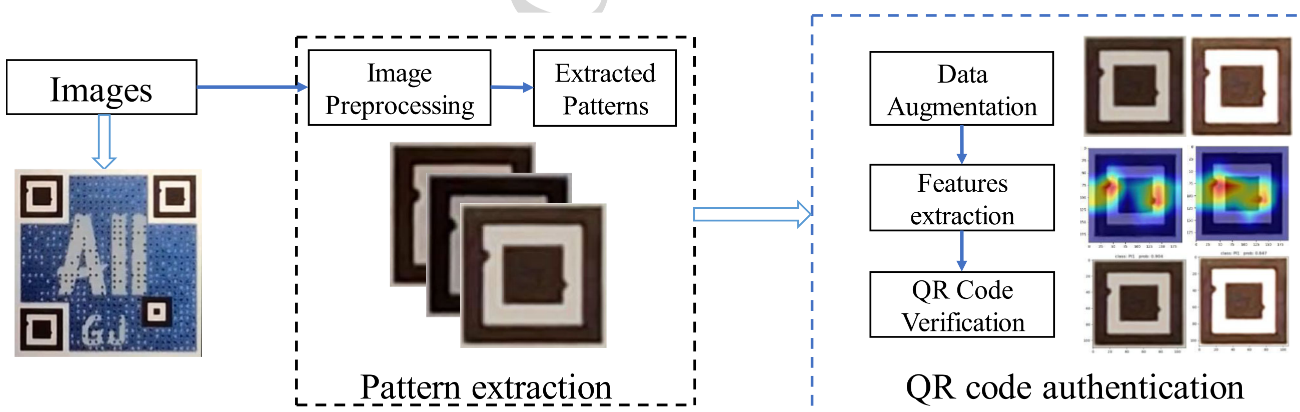


Fig. 5 Overall structure of the integrated QR code extraction and validation system, detailing the sequential processes from initial capture to final verification

Journal: **10489** MS: **6509** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2025/3/28** Pages: **22** Layout: **Large**

4.1.1 Modification of traditional adaptive thresholding

a. **Adaptive Window Size**
Traditional adaptive thresholding often uses a fixed window size for computing local statistics. This works well for images with consistent textural patterns and illumination levels. However, in the context of QR code extraction, this "one-size-fits-all" approach can prove inadequate. QR codes often appear at varying scales and might be embedded in backgrounds with diverse textural patterns or noise levels. Using a fixed window size can lead to suboptimal or erroneous thresholding in these scenarios.
To address the issues inherent in using a fixed window size, we propose a mathematically robust approach that allows for a dynamically adaptable window size, grounded in the statistics of the local neighborhood around each pixel. The mathematical formulation of the method can be defined as follows:
Let I be an image of dimensions $M \times N$, and let $p_{i,j}$ denote a pixel at the coordinates ((i,j)). The local variance $\sigma_{i,j}^2$ around this pixel is calculated as follows:

$$\sigma_{i,j}^2 = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I_{m,n} - \mu_{i,j})^2, \quad (1)$$

where $\mu_{i,j}$ represents the local mean of the pixel intensities within the window, and M, N are the dimensions of the window surrounding $p_{i,j}$.
The adaptive window size $W_{i,j}$ is then calculated as:

$$W_{i,j} = k \cdot \sigma_{i,j}, \quad (2)$$

where k is a proportionality constant that adjusts the impact of the local variance on the window size. This allows the window size to dynamically expand for areas with high variance, which could be indicative of edges, textures, or noise.

b. **Statistical Weighting**
Traditional adaptive thresholding often employs the mean value of a local neighborhood to set the pixel intensity threshold. While efficient, this approach lacks the granularity to handle complex scenarios, such as when QR codes are superimposed on textured or patterned backgrounds. Our study introduces statistical weighting into the thresholding equation to solve these issues. By considering higher-order statistical moments like skewness and kurtosis, our approach aims to capture more nuanced variations in pixel intensities. The relevant mathematical formulation of the proposed method can be defined as follows:
Let X be a random variable representing the pixel intensities within the adaptive window. Let μ and σ be the mean and standard deviation of X, respectively. Skewness (S) and kurtosis (K) of X are then defined as:

$$S = E\left[\left(\frac{(X-\mu)}{\sigma}\right)^3\right]. \quad (3)$$

$$K = E\left[\left(\frac{(X-\mu)}{\sigma}\right)^4\right] - 3. \quad (4)$$

The adaptive threshold $T_{i,j}$ for the pixel at coordinates (i, j) is computed using the following weighted formula:

$$T_{i,j} = \mu_{i,j} + \alpha \cdot S + \beta \cdot K, \quad (5)$$

Where α and β are weight parameters that control the influence of skewness and kurtosis, respectively, on the threshold value. These weights allow the method to adaptively adjust the thresholding, effectively capturing nuanced variations.

c. **Post-thresholding Refinement**
The thresholding process, while effective in isolating potential regions of interest, may produce pixel artifacts that can disrupt the distinct patterns of QR codes. These artifacts pose challenges in subsequent stages of QR code identification and decoding. To resolve this challenge, we introduce a post-thresholding refinement step that employs a Gaussian smoothing filter. The relevant mathematical formulation of the proposed post-thresholding refinement can be defined as follows:
The Gaussian smoothing filter is utilized to refine the pixels, which is mathematically defined as:

$$G_{i,j} = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (6)$$

where σ is the standard deviation that controls the spread of the Gaussian filter.
After applying the thresholding method, we obtain a thresholded image I_T. The refined image I' is then acquired by convolving I_T with the Gaussian filter G:

$$I' = I_T * G. \quad (7)$$

The proposed methods help smooth out minor pixel artifacts while preserving the essential boundaries that define the QR codes. The proposed post-thresholding refinement ensures that pixel artifacts are effectively eliminated, thereby improving the structural integrity of QR patterns.

4.1.2 Fine tuned edge detection

Edge detection serves as a foundational step in the image processing pipeline for QR code extraction. One of the most significant challenges in this context is the accurate identification of edges amidst varying conditions such as noise, uneven illumination, and complex backgrounds. To address these challenges, the Canny edge detection algorithm is introduced due to its robustness against noise and its ability to detect true edges with high accuracy. The detailed description of each step of the Canny edge detection is given below:

a. **Noise Reduction**
To reduce the influence of noise which can cause false edge detection, a Gaussian filter G is applied to the image:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{8}$$

where σ is the standard deviation.

b. **Gradient Computation**
The smoothed image is further processed using Sobel filters to compute the gradient magnitude G and direction θ for each pixel:

$$G = \sqrt{G_x^2 + G_y^2}, \theta = \arctan\left(\frac{G_y}{G_x}\right), \tag{9}$$

where G_x and G_y are the gradient magnitudes in the x and y directions, obtained using Sobel filters.

c. **Non-maximum Suppression**
Non-maximum suppression is utilized after gradient computation to ensure that the identified edges are thin by setting any pixel that is not a local maximum in its gradient direction to zero.

d. **Double Thresholding**
Canny edge detection employs two threshold values, T_{low} and T_{high}, to filter out gradients. Gradients are rejected when the pixel's gradient magnitude is less than T_{low}. Gradients are accepted when the magnitude of a pixel is higher than the T_{high} threshold.

e. **Edge Tracking by Hysteresis**
Pixels with gradient magnitudes between T_{low} and T_{high} are conditionally accepted as edges if they are connected to pixels with gradient magnitudes greater than T_{high}.

4.1.3 Contour extraction

After the use of edge detection, the subsequent and equally pivotal phase is contour extraction. This involves tracing the continuous boundaries detected by the edges, allowing us to segregate potential QR codes from other image components and backgrounds. This study introduces an enhanced contour extraction method that leverages hierarchical detection, filtering mechanisms, and orientation correction. The subsequent module introduced in the contour detection phase is described below in detail:

(a.) **Hierarchical Contour Detection**
Hierarchical contour detection extends beyond simple contour identification. It categorizes contours hierarchically based on their parent-child relationships, enhancing the capability to uniquely identify the characteristic nested square patterns of QR codes.
The contours can be represented as mathematical functions as follows:

$$C : [0, 1] \to \mathbb{R}^2, \quad C(t) = (x(t), y(t)). \tag{10}$$

In the hierarchical scenario, if a contour C_1 is entirely enclosed by another contour C_2, then C_1 is considered a child of C_2. This hierarchical nesting is pivotal for identifying the unique three-square pattern at the corners of QR codes.

(b.) **Contour Filtering**
This study introduces two primary filtering techniques: Aspect Ratio Filtering and Pattern Consistency for contour filtering.

1. Aspect Ratio Filtering:
The Aspect Ratio (AR) for a detected contour is computed as:

$$AR = \frac{\text{Height}}{\text{Width}}. \tag{11}$$

Contours with an aspect ratio significantly different from 1 (indicative of a square shape) are removed.

2. Pattern Consistency:
QR codes possess three large squares at their corners, allowing for pattern consistency checks within contours to eliminate false positives.

(c.) **Orientation Detection**
Orientation is key to the accurate decoding of QR codes. As images can capture QR codes in various orientations, a robust methodology to detect and correct these orientations is crucial. The moment-based technique was introduced to detect orientation.
Central moments μ_{pq} are used to compute the orientation of a contour and are defined as:

$$\mu_{pq} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q f(x, y), \tag{12}$$

Where (x_c, y_c) is the centroid of the shape, and $f(x, y)$ is the image intensity at the coordinates (x, y).

Journal: **10489** MS: **6509** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2025/3/28** Pages: **22** Layout: **Large**

Using central moments, the orientation θ can be computed as:

$$\theta = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{13}$$

This angle θ provides the angular deviation from the standard orientation. The QR code is then rotated by this angle to ensure it is optimally oriented for decoding.

4.2 Deep learning based QR code verification system

The classification of QR codes faces resource-intensive processing, background interference, and the critical need for high accuracy. To address these challenges, a solution has been developed that involves the use of the lightweight ShuffleNet v2 network, enhanced through transfer learning and optimized activation functions. This novel approach offers an end-to-end QR verification/classification model that strikes a balance between efficiency and accuracy, as depicted in Fig. 6.

It starts by forming a foundational feature extraction network using the ShuffleNet v2 framework, which serves as the backbone. To enhance the model's initial state for training, the weights of this backbone network are initialized through transfer learning. This strategic step fine-tunes the model's starting point, bolstering the prominence of valuable features and downplaying less pertinent ones.

Furthermore, the choice of the Rectified Linear Unit (ReLU) activation function is deliberate. By incorporating ReLU, the model excels in extracting spatial context features from the data. This capability empowers the model to discern intricate patterns and relationships within the input images. An additional advantage of ReLU is its ability to prevent neurons from being deactivated when input data contain negative values, thus ensuring a more consistent and effective training process.

The construction of the model revolves around harnessing the strengths of ShuffleNet v2, augmenting its performance through transfer learning, and optimizing the feature extraction process using the ReLU activation function.

ShuffleNet v2

The evolution of convolutional neural network (CNN) architectures has ushered in remarkable breakthroughs, redefining the landscape of efficiency and accuracy. This chapter presents ShuffleNet v2 [41], an evolutionary advance beyond its precursor, ShuffleNet v1, introduced by MEGVII. Guided by four design principles and propelled by the innovative channel shuffle mechanism, ShuffleNet v2 represents a significant change in CNN design. It outshines its predecessors in accuracy while upholding computational efficiency.

Rooted in the ethos of efficiency and performance, ShuffleNet v2 introduces the concept of channel shuffle, ingeniously overcoming the limitations posed by grouped convolution. Grouped convolution, pioneered by Krizhevsky et al. [42] and Zhang et al. [43], economizes computational resources by focusing convolution kernels on specific channel groups. However, this efficiency compromises intergroup information exchange, hindering feature expressiveness. Inter-channel shuffle, proposed by ShuffleNet [44], is a simple yet transformative stratagem that disrupts the output features of previously grouped convolutions in the channel dimension.

Four distinctive characteristics served as the foundation for ShuffleNet v2's design and development, resulting in the cell structure seen in Fig. 7. This framework depends on DW convolution, which stands for depthwise convolution, and channel separation, a method that divides input features into discrete parts [45]. These components are expertly put together to form the fundamental building block of

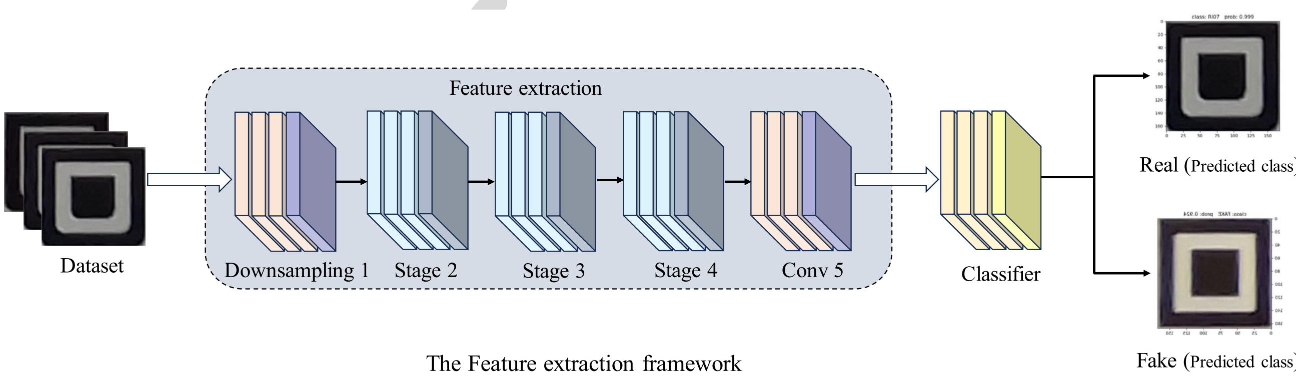


Fig. 6 End-to-end QR classification using the optimized ShuffleNet v2 network, highlighting a balance between efficiency and accuracy

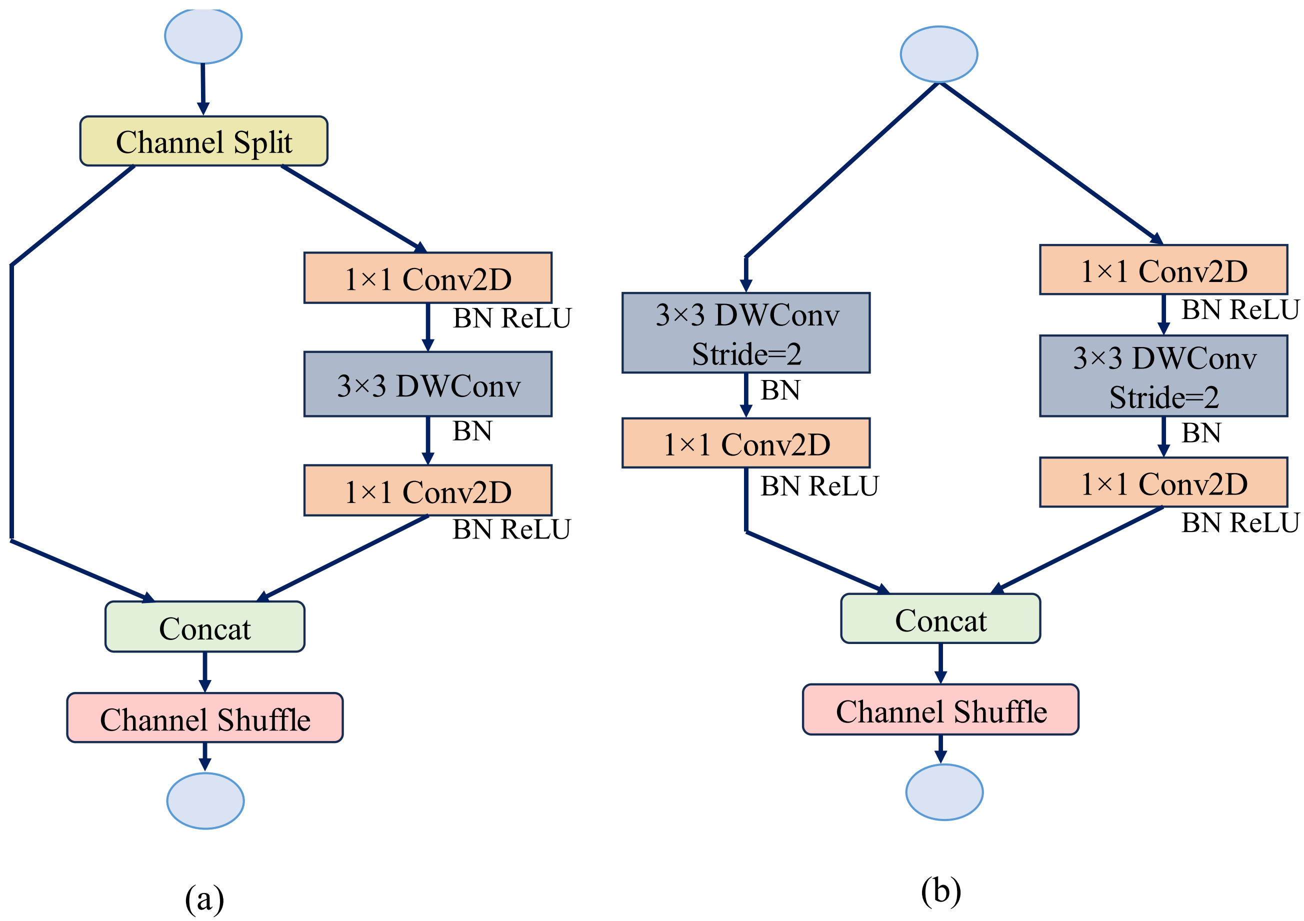


Fig. 7 ShuffleNet v2 cell structure, showcases its design based on channel separation and depthwise convolution

ShuffleNet v2, each meticulously aligned with four guiding principles that underpin its design stance. These guiding principles include the following:

1. **Memory efficiency** One fundamental principle in the search for efficiency is the tactical control of memory access costs. To establish equilibrium, the input and output channel counts inside the convolutional layers are purposefully aligned. The input feature dimensions for a 1×1 convolution span c_i channels and $h \times w$ spatial dimensions, while c_o denotes the number of output channels. The result of this orchestration, which aims to reduce memory access costs, is a crucial formula where F, which stands for FLOPs (Floating-Point Operations), is related to the overall scenario, and where h, w, x_i, x_o, and c_o are, respectively, the height, width, number of input channels, and the number of output channels.
2. **Optimizing efficiency via controlled group convolutions** Strategic efficiency is further increased by consciously using fewer group convolutions, as excessive use can result in an increase in memory access costs. The term "g," which refers to the number of groups in a group convolution, is crucial in this situation. These can be defined as follows:

$$F = \frac{h \cdot w \cdot x_i \cdot x_o}{g} \tag{14}$$

$$\mathrm{MAC} = h \cdot w \cdot (x_i + x_o) + \frac{x_i \cdot x_o}{g} = h \cdot w \cdot x_i + \frac{Fg}{x_i + F/hw} \tag{15}$$

3. **Reducing network branches** Reducing the number of network branches increases efficiency. A network can lag if it has too many branches. For instance, several multi-branch structures are utilized as the fundamental building elements of the Inception architecture. However, we must be cautious, as having too many of these branches can dramatically reduce the computer's capacity for parallel processing.
4. **Streamlining Tensor Operations for Increased Efficiency** Reducing the number of tensor operations is one technique to improve efficiency. However, even straight-

Journal: **10489** MS: **6509** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2025/3/28** Pages: **22** Layout: **Large**

forward functions like ReLU and adding features can place a significant burden on the Multiply-Accumulate (MAC) resources. Therefore, it's crucial to optimize even these small processes for greater effectiveness.

5 Experimental result and discussion

5.1 Experimental parameters

The majority of the Python 3.7 code used to create the QR verification framework described in this proposal was run on an Ubuntu 20.04 machine. This machine was powered by two Nvidia Tesla V100 GPUs, each with 12 gigabytes of RAM. The model relied on the ShuffleNetV2 architecture, implemented using PyTorch 2.0.0-a highly recognized open-source deep learning framework known for its adaptability and robustness in fundamental image classification and QR verification tasks.

We used a training pipeline for a neural network, utilizing the Cross-Entropy Loss function, which measures the dissimilarity between predicted class probabilities and actual labels for multi-class classification tasks. To optimize our model, we employed Stochastic Gradient Descent (SGD) with momentum and weight decay. The learning rate (lr) was set to 0.01, controlling the step size during weight updates, while the momentum factor enhanced training stability. Additionally, a learning rate scheduler employing a cosine annealing strategy helped fine-tune the lr throughout training. Hyperparameters included the number of classes (24), the number of training epochs (50), batch size (8), and device choice (GPU).

5.2 Evaluation metrics

In our model evaluation, we utilized recognized metrics for evaluating multiclass classification. We carefully identified the occurrences of true positives (TP), true negatives (TN), false negatives (FN), and false positives (FP) for each classification test. The average classification accuracy (A), average recall (R), average precision (P), and the F-1 score (F-1) were then calculated using these fundamental data.

The formula for the Average Classification Accuracy (A), frequently considered a fundamental indicator of model performance, is the sum of the combined results of the TP, TN, FP, and FN. It reflects the overall accuracy of our classification predictions and provides insightful information about the model's performance.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

Average Recall (R), an essential metric in multiclass classification, assesses how well our algorithm recognizes positive cases. It measures the relationship between true positives (TP) and the total of true positives and false negatives (FN), reflecting the model's ability to incorporate relevant information.

$$R = \frac{TP}{TP + FN} \tag{17}$$

An important performance statistic, called Average Precision (P), evaluates how well the model predicts positive events. The accuracy of our model's predictions is calculated as the ratio of true positives (TP) to the sum of true positives and false positives (FP).

$$P = \frac{TP}{TP + FP} \tag{18}$$

The F1 Score (F1), a comprehensive evaluation of classification performance, harmoniously combines recall and precision. It is calculated as the harmonic mean of recall (R) and precision (P), providing a balanced measure of both metrics. The F1 Score is a crucial parameter in classification assessments, offering an accurate evaluation of our model's ability to achieve both high precision and recall simultaneously.

$$F = \frac{2 \cdot (R \cdot P)}{R + P} \tag{19}$$

5.3 Quantitative analysis

We conducted experiments using thirteen different CNN architectures to facilitate a comprehensive comparison. Throughout the experiments, the training and validation datasets were randomly distributed in a ratio of 80:20. The results from these experiments are summarized in Table 2. ShuffleNetV2 [41] emerged as the leader after 50 training iterations, achieving the highest average validation accuracy of 99.99%, closely followed by ResNet101 [46] at 99.95% and DenseNet121 [47] at 99.93%. The accuracy of the remaining CNN architectures ranged from 91.89% to 99.0%. Furthermore, state-of-the-art models such as ConvNeXt [48] and ConvNeXt-v2 [49] achieved accuracies of 98.12% and 98.85%, respectively. Similarly, MobileViT [50] and Mobile-ViT-v2 [51] attained accuracies of 99.28% and 99.45%, respectively. In comparison, our model demonstrated superior performance across all metrics.

Additionally, we tested the model's performance using accuracy, recall, F1-score, and AUC as evaluation measures, which gave us a more thorough understanding of its capabilities. Interestingly, ShuffleNetV2 exceeded the other models in terms of accuracy 99.76%, recall 99.76%, and F1-score

Table 2 Comparative analysis of the evaluation metrics of various deep learning models for the QR code verification task

Model	Accuracy	Recall	Precision	F-1 Score
ResNet-101 [46]	99.95	99.30	99.31	95.24
ResNet-50 [46]	99.95	99.45	99.45	99.44
DenseNet-121 [47]	99.93	99.04	99.05	99.03
MobileNet-v3 [52]	91.89	91.92	91.91	91.87
MobileNet-v2 [53]	98.27	98.12	97.95	97.91
RegNetx-800 mf [54]	99.89	98.62	98.62	98.61
RegNety-800 mf [54]	99.91	98.87	98.88	98.86
ConvNeXt [48]	98.12	81.86	84.94	82.16
ConvNeXt-v2 [49]	98.85	87.64	89.73	87.21
Mobile-VIT [50]	99.28	98.12	99.13	99.07
Mobile-VIT-v2 [51]	99.45	98.84	99.25	99.18
VIT [55]	98.32	85.40	82.56	74.80
ShuffleNet-v2 [41]	99.99	99.76	99.76	99.75

The models include ResNet-101, ResNet-50, DenseNet-121, MobileNet-v3, MobileNet-v2, RegNetx-800mf, RegNety-800mf, ConvNext, Mobile-ViT, ViT, and ShuffleNet-v2. ShuffleNetV2 achieved the highest performance across most metrics, including a validation accuracy of 99.99%, recall of 99.76%, and F1-score of 99.75%. ResNet101 was the second-best performer with an accuracy of 99.95%. ShuffleNetV2 significantly enhances model performance compared to traditional and recent state-of-the-art approaches

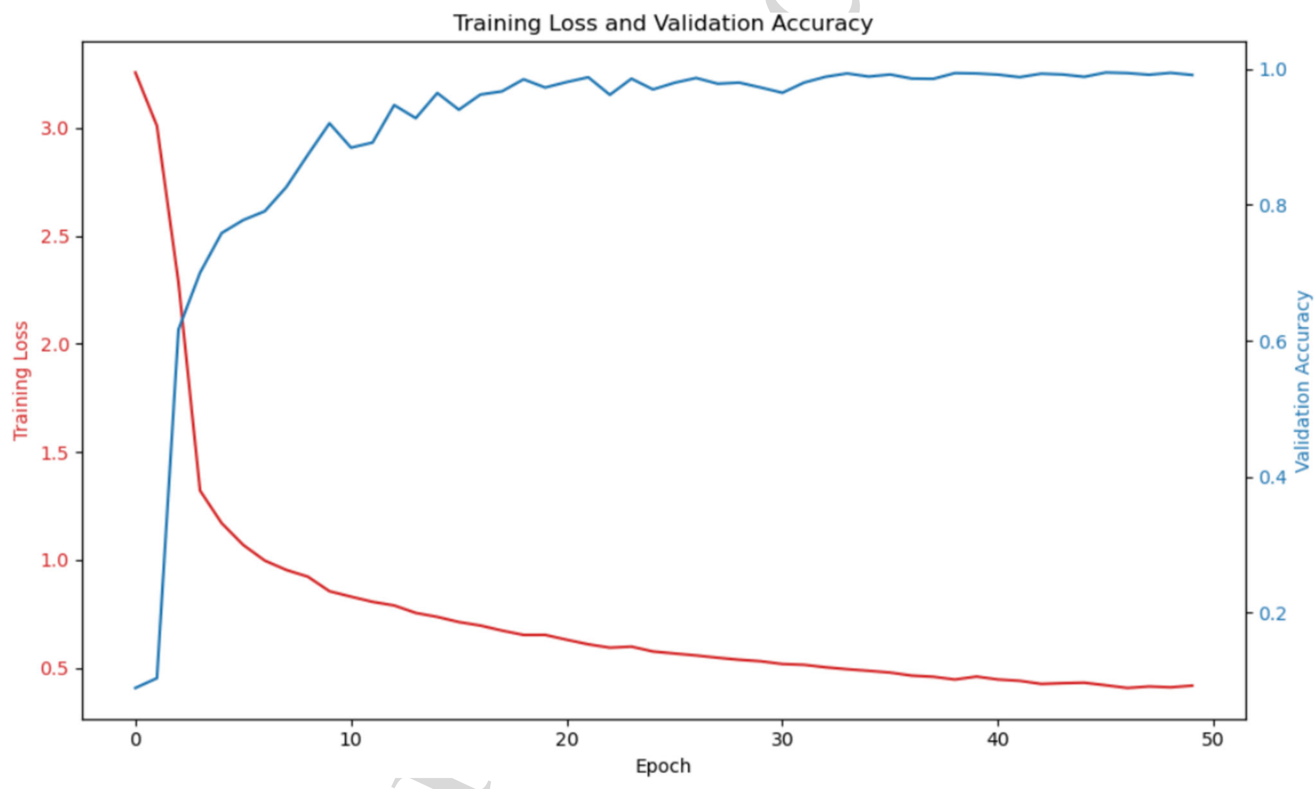


Fig. 8 Training loss and validation accuracy curves for 50 epochs using the ShuffleNetV2 model. The figure highlights the stabilization of validation accuracy and training loss, demonstrating the model's convergence. The highest accuracy of 99.94% was achieved at the 45th epoch, indicating optimal model performance

Journal: **10489** MS: **6509** TYPESET DISK LE CP Disp.:**2025/3/28** Pages: **22** Layout: **Large**

99.75%. In terms of these evaluation metrics, ResNet101 took second place on the list.

These results lead us to the conclusion that, with the MobileNet [52] architecture, increasing the amount of data through data augmentation has very little impact on improving the accuracy of QR pattern classification. On the other hand, methods such as separable depth convolutions and efficient channel-wise operations support better training and performance of the deep neural network, as shown in Table 2.

A convolutional neural network (CNN) model, built on the ShuffleNetV2 architecture, has been trained to classify a diverse collection of QR pattern images from various time periods.

Figure 8 presents the training and validation performance of the ShuffleNetV2 model over 50 epochs. The training loss decreased steadily, stabilizing around the 46th epoch, while the validation accuracy plateaued by the 40th epoch and achieved its highest value of 99.94% at the 45th epoch. These trends indicate effective training and convergence of the model, confirming its suitability for low-latency QR code varification tasks.

We have chosen ShuffleNetV2 as the final classification algorithm for our QR verification system because of its excellent classification performance and computational effectiveness. The confusion matrix plot for the ShuffleNetV2 architecture employed in the system can be seen in Fig. 9. In this plot, the columns correspond to the true label classes (Target Class) and the rows to the predicted label classes (Output Class). The off-diagonal cells in Fig. 9 show the number of validation samples for QR code patterns that were incorrectly classified, while the diagonal cells show the number of correctly classified validation samples for similar patterns.

In Table 3, we present a comprehensive comparison of the processing time between our proposed framework and other widely accepted classification methods for QR code images. Specifically, we calculate the total processing time for each QR code image. Our results show that our proposed algorithm achieves an impressive processing time of only 0.08 seconds, slightly outperforming its counterparts.

We also evaluate the parameter estimates for these methods. Table 3 reveals that our proposed framework

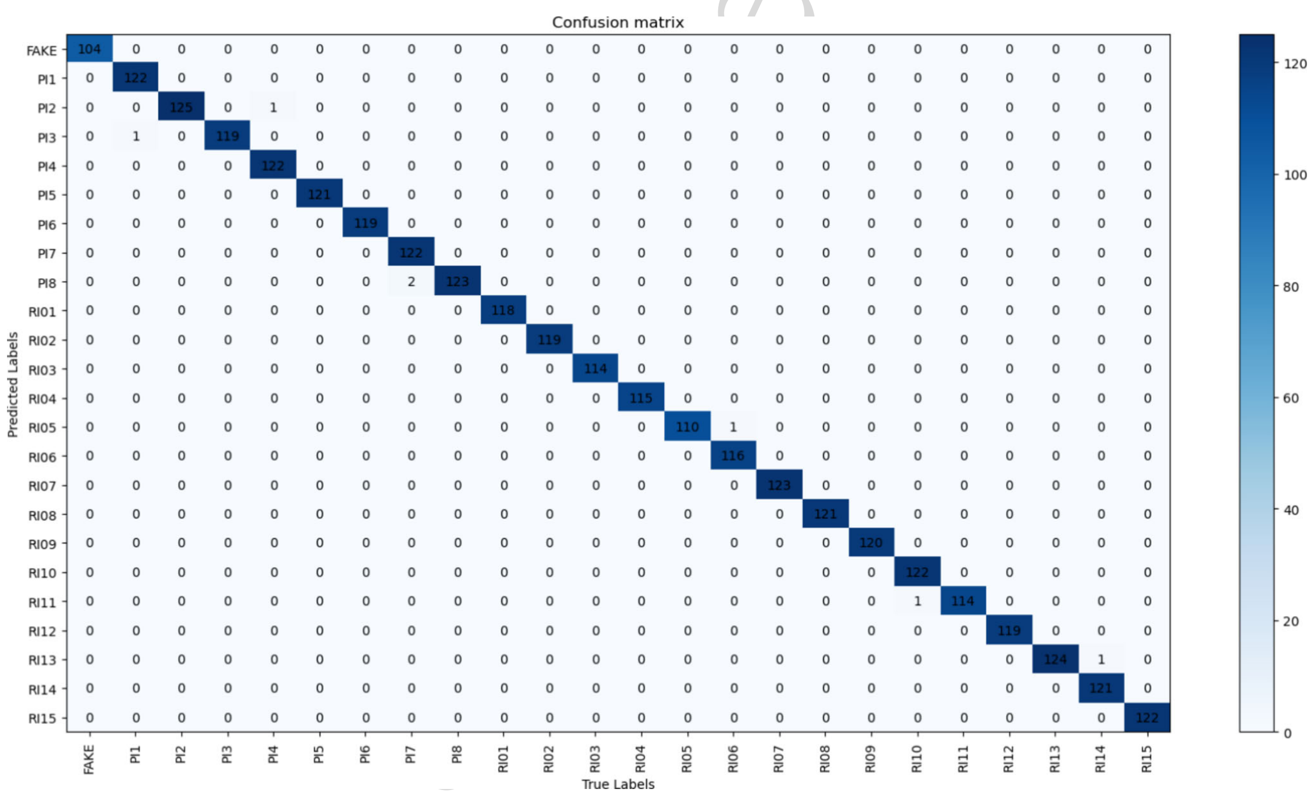


Fig. 9 Confusion matrix showcasing the classification performance of the ShuffleNetV2 model on the test dataset for the QR verification system. The columns represent the true label classes (Target Class), and the rows represent the predicted label classes (Output Class). Diagonal cells indicate the number of correctly classified validation samples for each QR code pattern, while off-diagonal cells display the number of incorrectly classified samples. This visualization highlights the exceptional classification performance and computational effectiveness of the ShuffleNetV2 architecture in our system.

Table 3 Comparative analysis of different algorithms based on execution time (in seconds) and parameter size (in MB)

Algorithm	Time (s)	Params size (MB)
RegNet800mf [54]	0.18	21.03
ResNet50 [46]	0.53	89.86
ResNet101 [46]	0.95	162.31
EfficientNet_b0 [56]	0.17	15.40
Convnext [48]	0.91	188.73
Mobile-VIT [50]	0.11	3.66
VIT [55]	1.45	329.62
ShuffleNetV2 [41]	0.08	4.88

ShuffleNetV2 demonstrates the fastest processing time of 0.08 seconds and the smallest parameter size of 4.88 MB, making it ideal for real-time mobile QR verification apps. Other methods, including RegNet800mf, ResNet50, ResNet101, EfficientNet-b0, Convnext, and VIT, have larger parameter sizes and slower processing times

is remarkably efficient, with a modest parameter size of 4.88 MB. In contrast, RegNet800mf [54] uses 21.03 MB, ResNet50 [46] uses 89.86 MB, ResNet101 [46] uses 162.31 MB, EfficientNet-B0 [56] uses 15.40 MB, ConvNext [48] uses 188.73 MB, and Mobile-ViT [50] and ViT [55] use 3.66 MB and 329.62 MB, respectively, for parameter extraction. The combination of reduced processing time and compact parameter size positions our proposed framework as an ideal choice to deploy QR verification systems for real-time mobile applications.

5.3.1 Hyperparameter optimization

In order to achieve the best possible accuracy for a classification model, hyperparameters are crucial. The learning rate and the selection of the optimization technique are the most critical of these hyperparameters. As shown in previous studies [57, 58], an insufficiently adapted learning rate could cause erratic loss variations and a delayed convergence pace. To choose the best hyperparameters for the model we propose, SGD stands out among the many optimization techniques. The impact of various hyperparameters on model performance is clearly shown in Fig. 10. We used the SGD optimizer to train the model proposed in our studies. To explore the effects of different learning rates, we selected three values: 0.01, 0.005, and 0.001 for the optimization setting. Our goal was to compare the validation accuracy and training loss of these settings to choose the most suitable learning rate for future analysis.

With the SGD optimizer, especially with a learning rate of 0.01 and momentum of 0.9, the model achieved its highest accuracy. Based on these findings, we determined that the best hyperparameter configuration for the model we proposed was a learning rate of 0.01 and the SGD optimizer. This fine hyperparameter tuning process sets the stage for maximum model performance, robustness in subsequent evaluations and highlight the superiority of a learning rate of 0.01, which achieves the best balance between accuracy and convergence speed, demonstrating the importance of fine-tuning hyperparameters to ensure optimal performance.

5.3.2 Ablation study

To evaluate the performance of ShuffleNetV2 under different configurations, we conducted an ablation study on the QR pattern dataset, varying input sizes, network depth, activation functions, and optimization techniques. The results, summarized in Table 4, demonstrate that ShuffleNetV2 achieves consistent and high performance across these parameters. Notably, the input size of 224×224 emerges as the optimal configuration, yielding an accuracy of 99.99%. Similarly, the standard ShuffleNetV2 network depth achieves the best results, while deeper and shallower variants show slight performance variations. Regarding activation functions, ReLU and Leaky ReLU both produce high accuracy, with ReLU slightly outperforming. Finally, optimization techniques reveal that SGD with Momentum delivers superior performance compared to Adam. These findings highlight the robustness and adaptability of ShuffleNetV2 across different parameter settings.

5.4 Qualitative analysis

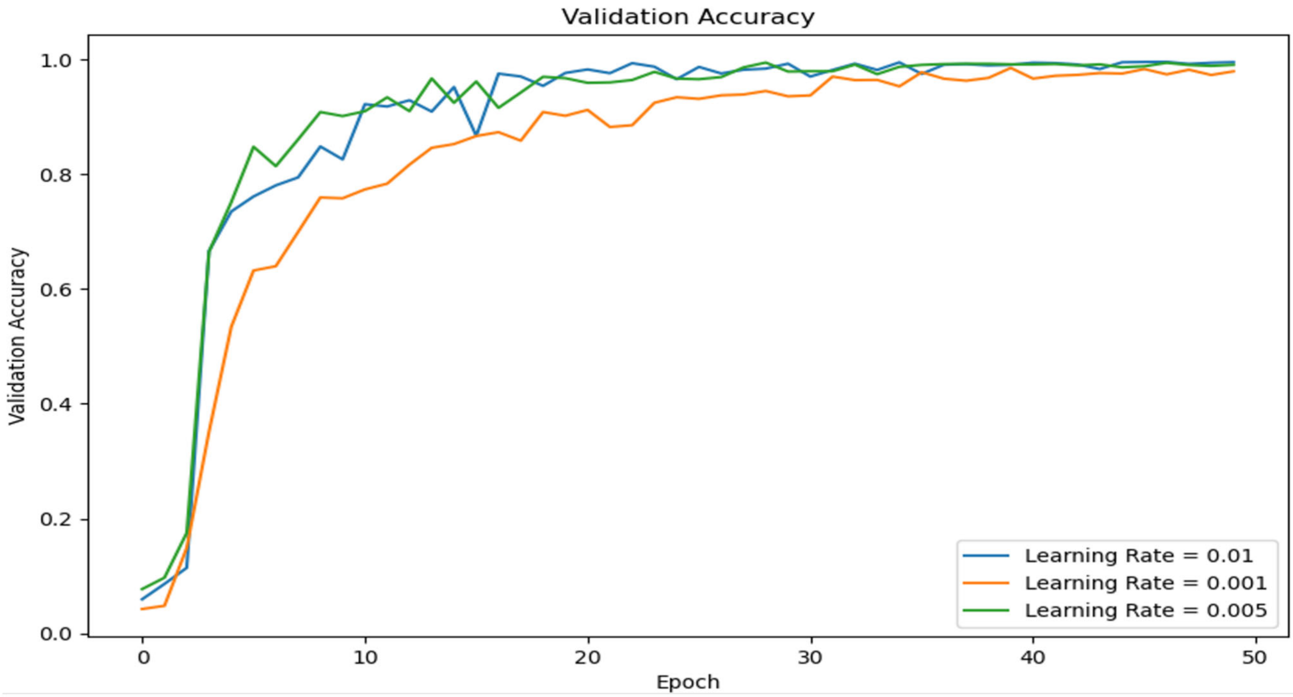
In Fig. 11, we show how well our model performs when tested against QR code images related to sample authentication. The findings demonstrate its remarkable capacity to classify QR codes efficiently, enhancing its reliability for authentication tasks. The robustness and accuracy of the model in classifying QR codes are evidenced by its consistent achievement of an average prediction score of over 99%, making it the best choice for authentication purposes.

To ensure that our model can handle real-world scenarios effectively, we need to evaluate its performance under various types of noise, which is a significant concern. Therefore, we evaluate our model's ability to perform in noisy conditions before applying it for real-time QR code verification.

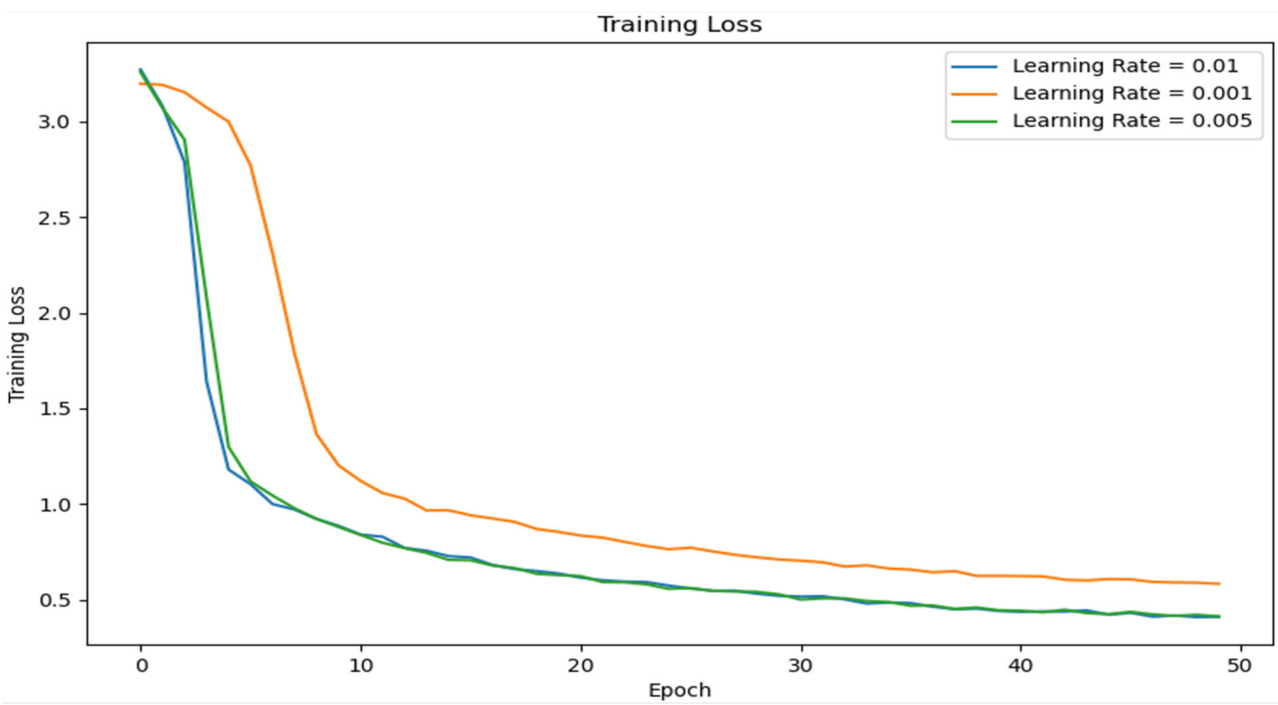
Our evaluation focuses on four common types of noise that frequently appear in QR code images: Gaussian Noise, Blur, Lighting Variations, and Random Printed Noise. These types of noise represent the challenges we may encounter due to the environment and equipment. Our objective is to prove that even in the presence of these possible sources of interference, our model can reliably and accurately distinguish real QR codes from false ones.

The prediction accuracy of the proposed model applied to images with Gaussian noise is shown in Fig. 12. The original picture, added Gaussian noise, and input into the classifier for prediction are shown in Fig. 12(a). We added 20

Journal: 10489 MS: 6509 TYPESET DISK LE CP Disp.:2025/3/28 Pages: 22 Layout: Large



(a)



(b)

Fig. 10 The comparison of validation accuracy and training loss for the ShuffleNetV2-based CNN model under different learning rates. The learning rates tested were 0.01, 0.005, and 0.001, using the stochastic gradient descent (SGD) optimizer. The results demonstrate the significance of hyperparameter tuning, as inadequate learning rates can lead to inconsistent loss variations and slower convergence

Journal: **10489** MS: **6509** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2025/3/28** Pages: **22** Layout: **Large**

Table 4 Ablation study on model parameters, including modifications in input size, network depth, activation functions, and optimization techniques

Parameter	Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Input size	224×224	99.99	99.76	99.76	99.75
	227×227	99.60	99.50	99.49	99.48
	256×256	99.53	99.30	99.32	99.31
Network Depth	Standard ShuffleNetV2 (baseline)	99.99	99.76	99.76	99.75
	Deeper Variant	99.85	99.70	99.71	99.69
	Shallower Variant	99.40	99.20	99.15	99.10
Activation Function	ReLU	99.99	99.76	99.76	99.75
	Leaky ReLU	99.80	99.60	99.58	99.59
Optimization Technique	SGD with Momentum	99.99	99.76	99.76	99.75
	Adam	99.75	99.50	99.49	99.50

The results emphasize the performance of ShuffleNetV2 and its variants, evaluated using accuracy, precision, recall, and F1 score metrics

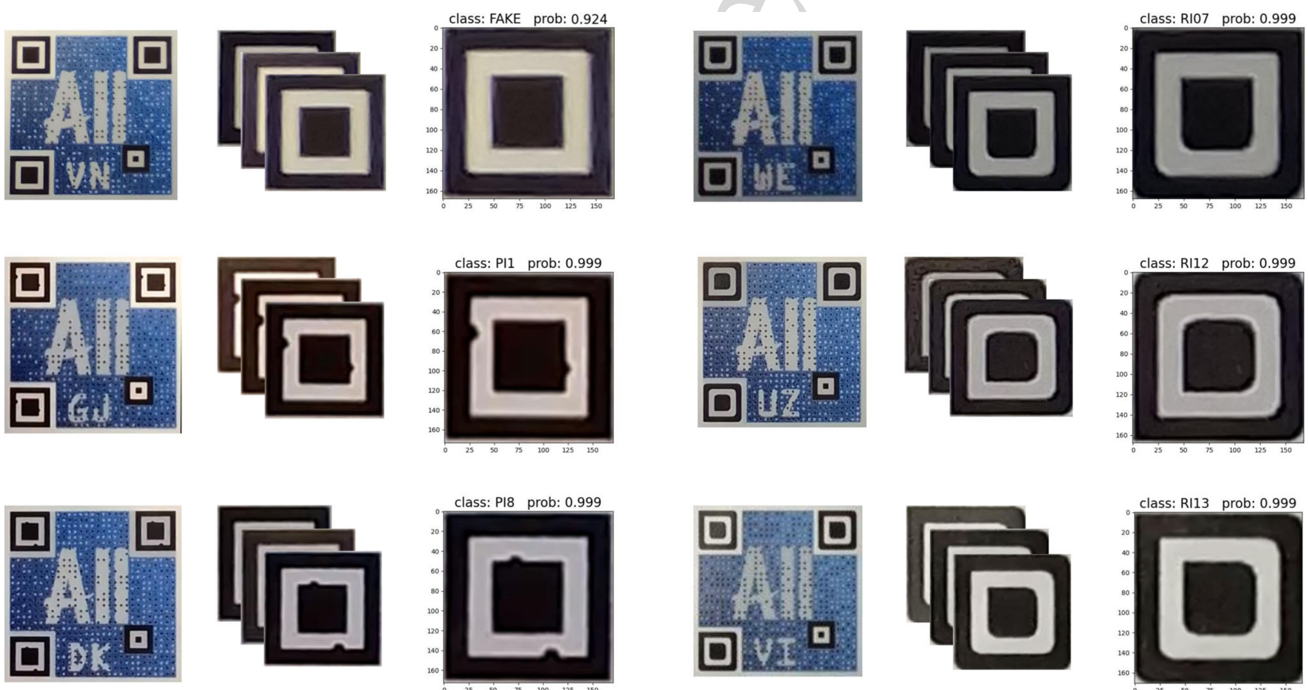


Fig. 11 Visualization of predicted QR code pattern classes using the ShuffleNetV2 model. It demonstrates the model's outstanding performance in classifying QR code images related to sample authentication, consistently achieving an average prediction score of over 99%. The high reliability and precision of the ShuffleNetV2 model make it an ideal choice for QR code authentication tasks

Journal: 10489 MS: 6509 TYPESET DISK LE CP Disp.:2025/3/28 Pages: 22 Layout: Large

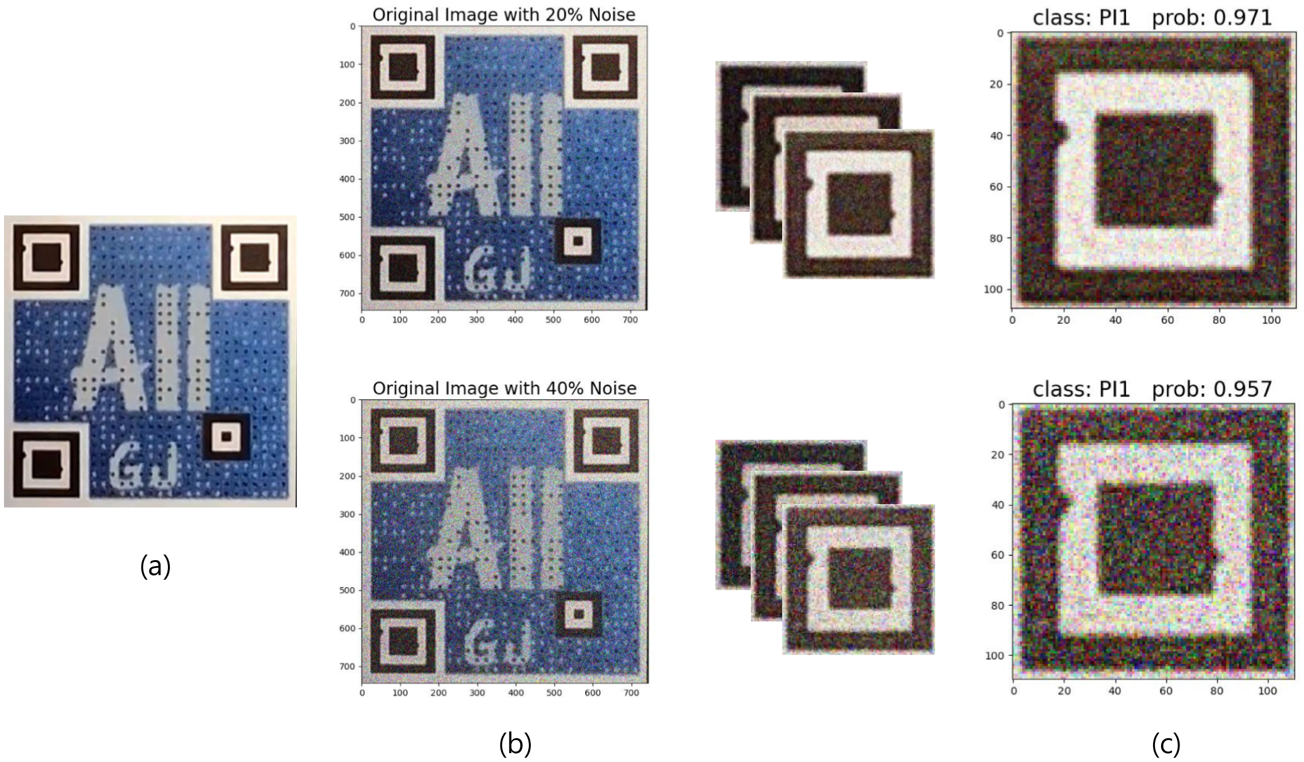


Fig. 12 Performance evaluation of the proposed model on images with Gaussian noise. The original image, along with images with 20% and 40% added Gaussian noise, are shown in this figure. The proposed model consistently achieves high prediction accuracy with prediction scores exceeding 95% even at 40% noise intensity, demonstrating the model's robustness and accuracy in noisy conditions

The proposed model consistently predicts the class with high prediction accuracy. In Fig. 12(c), it is evident that the average prediction score for Gaussian noise images exceeded 95%, indicating the model's robustness and accuracy in classifying patterns even in the presence of noise. In Fig. 13, we present the prediction accuracy of the proposed model applied to images with random blur. Notably, our proposed classification model achieved an impressive accuracy rate of more than 97

The high prediction score for blurred images can be attributed to the fact that the distribution of QR pattern symbols remained largely unchanged even when the image was blurred. Consequently, our proposed model consistently classified the correct class for constrained noisy QR code images. This demonstration underscores the versatility of the model and its potential applicability in various constrained environments for QR code classification and authentication.

Additionally, we tested the proposed model against a number of lighting variations that are frequently present in real-life scenarios involving QR code pictures. The expected results of the proposed model on images subjected to lighting variations are shown in Fig. 14.

For the same image, lighting variations were intentionally added, and the proposed model predicted every pattern successfully. The accuracy ranged from 90.4% for the highest accuracy to 84.7% for the lowest accuracy, as shown in Fig. 14(c).

Lastly, we evaluated our proposed QR code authentication model against Printed Noise, a common real-world scenario involving QR code images. In Fig. 15, you can see the model's predictions on images with Printed Noise intentionally introduced.

In Fig. 15(c), the proposed model correctly predicted every image, with an accuracy rate ranging from 89.3% to 98.4%.

6 Discussion

6.1 Comparison with existing methodology

We evaluate the performance of our proposed method against several existing methodologies in the field of QR code validation. We compare our approach with the Siamese network [59], Combined (Grab Cut + Image Splicing + SIFT + Optical Character Recognition) [60], AlexNet, and

Journal: 10489 MS: 6509 TYPESET DISK LE CP Disp.:2025/3/28 Pages: 22 Layout: Large

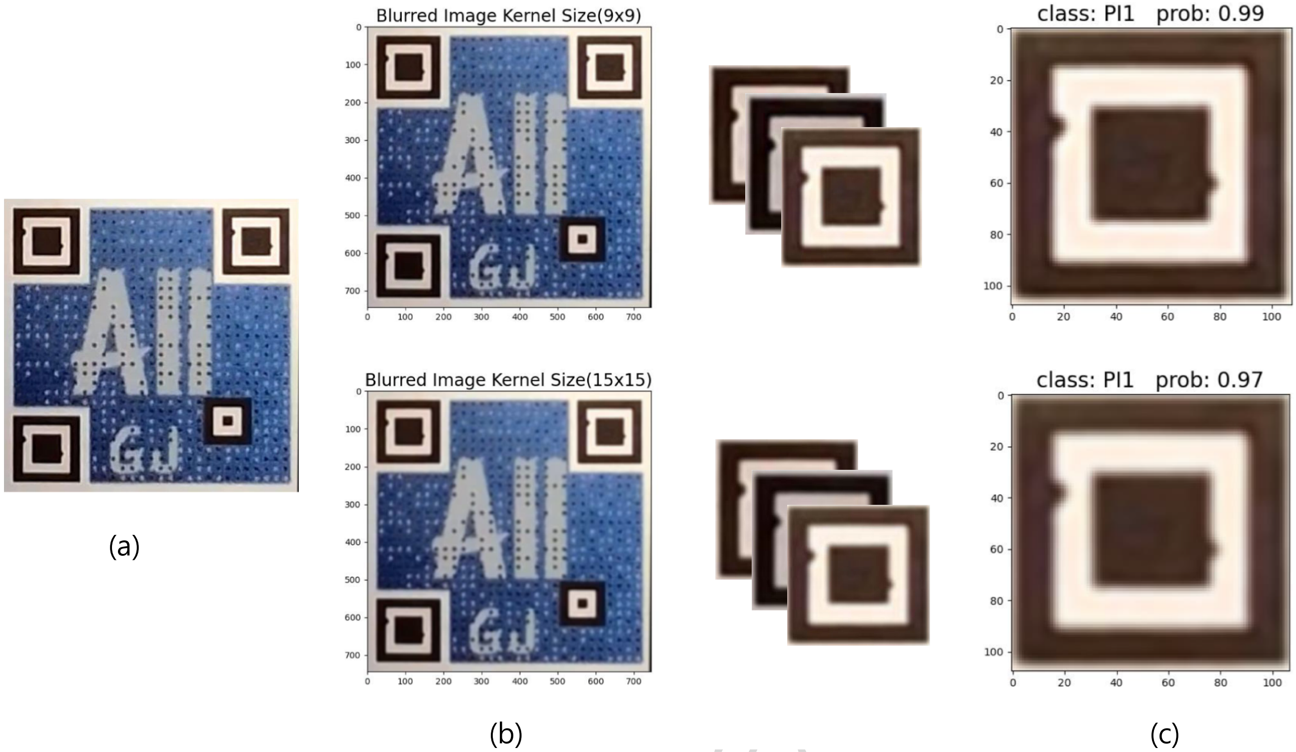


Fig. 13 Evaluation of the model's prediction accuracy on blurred images using kernel sizes ranging from 9×9 to 15×15. The proposed classification model consistently achieved accuracy rates greater than 97% for QR code classification, demonstrating its robustness. The high prediction scores indicate that the distribution of QR pattern symbols remains largely unaffected by blurring. This highlights the model's versatility and potential applicability in various constrained environments for QR code classification and authentication

ResNet18 [61] based on their dataset sizes and achieved accuracies, as shown in Table 5.

Our method achieves an excellent accuracy of 99.99% with a dataset size of 28,523, surpassing existing methodologies. While the Siamese network demonstrates promising results at 98% accuracy, our approach significantly outperforms it. Despite the Combined method's integration of various techniques, it achieves an average accuracy of 85.25%, highlighting the effectiveness of our proposed method. Furthermore, compared to traditional deep learning architectures like AlexNet and ResNet18, our method demonstrates superior accuracy, emphasizing its practical applicability. These results affirm the effectiveness and robustness of our proposed methodology.

6.2 Advantages, limitations and future directions

The existing study on QR code extraction and verification in dynamic environments has been limited in providing an efficient framework that includes both precise QR extraction techniques and robust verification methods. This research aims to address this gap by proposing a novel technique with a deep learning-based verification approach that differentiates itself from previous methods. A significant contribution of this study is its outstanding performance in both processing speed and verification accuracy, achieving a notable processing time of 0.08 seconds. The methodologies and experimental results demonstrated qualitative and quantitative agreement, establishing the reliability of the findings.

Previous frameworks mostly relied on traditional methods such as thresholding, dilation, and contour detection for QR code extraction [60]. However, these methods often fail to deliver satisfactory accuracy and processing speed. One of the leading challenges remains the impact of document image quality on QR code extraction effectiveness. Variations in lighting conditions, viewing angles, and image resolutions can substantially affect the accuracy of QR code detection and pattern extraction. Furthermore, traditional filtering and bounding box techniques may not consistently identify the QR code region, leading to false positives and false negatives that compromise the reliability of pattern extraction and verification processes.

To overcome these challenges, our study introduces a comprehensive approach that utilizes enhanced adaptive

Journal: 10489 MS: 6509 ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:2025/3/28 Pages: 22 Layout: Large

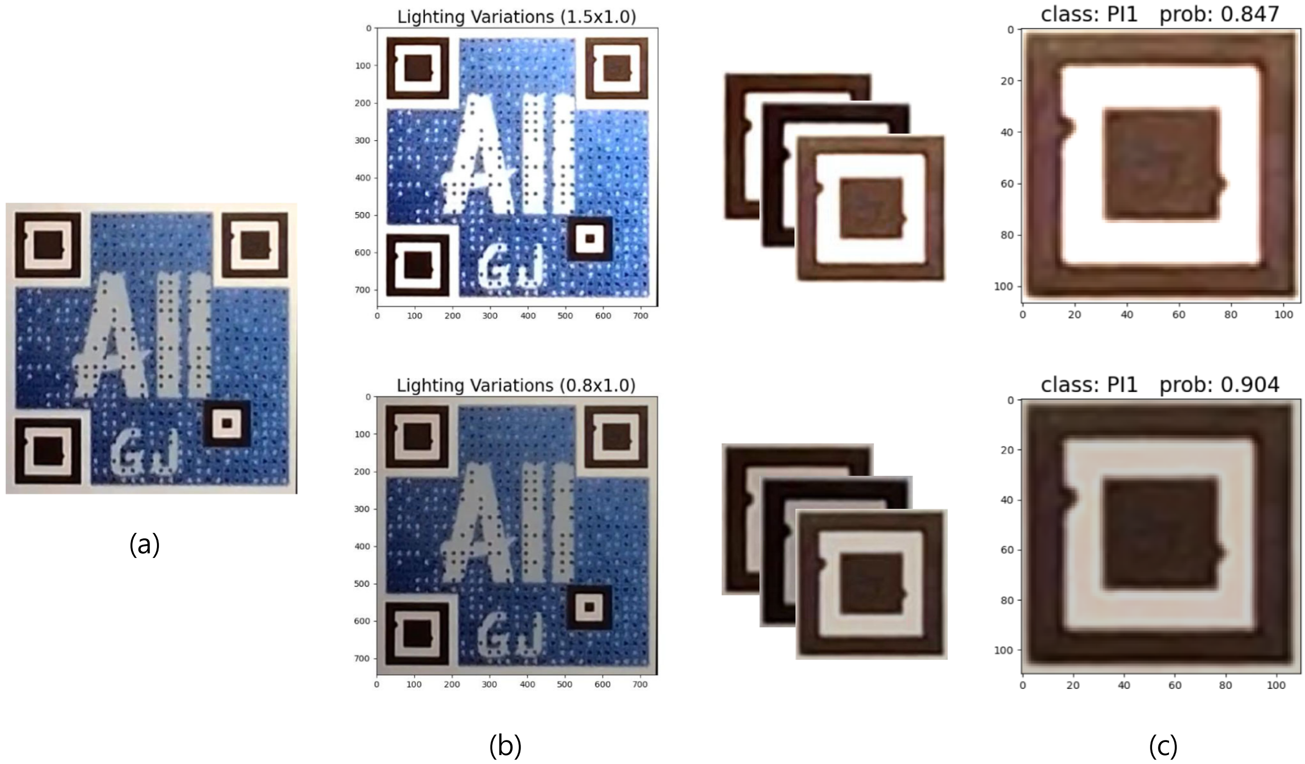


Fig. 14 Assessment of the model's performance on images with intentional lighting variations, illustrating accuracy rates ranging from 84.7% to 90.4%. The proposed model was tested against various lighting conditions commonly encountered in real-life scenarios involving QR code images. Despite these variations, the model consistently predicted the correct patterns, demonstrating its robustness and reliability under different lighting conditions

thresholding for QR code extraction and integrates a deep learning framework designed for robust QR code verification. We trained various state-of-the-art classification models, including ShuffleNetV2, ResNet, MobileNet, RegNetx, and DenseNet, on the proposed QR pattern dataset. Among these models, the ShuffleNetV2 model showed the highest accuracy of 99.99%, demonstrating its precise classification capabilities for QR pattern images.

Table 5 comparison of several QR Code Validation Methodologies, highlighting the dataset sizes and accuracy levels attained by Siamese network, Grab Cut + Image Splicing + SIFT + Optical Character Recognition), AlexNet, ResNet18, and our proposed approach

Method	Dataset Size	Accuracy
Siamese network [59]	5000	98%
Combined (GrabCut+Image Splicing+SIFT+Optical Character Recognition) [60]		85.25%
AlexNet [61]	2640	95.04%
ResNet18 [61]	2640	99.96%
Ours	28523	99.99%

Our proposed framework showcases its robustness in handling the intricacies of real-world scenarios, delivering impressive prediction accuracy rates ranging from 90.04% to 99.00% for complex and varied environments. By combining advanced extraction techniques with deep learning-based verification, our approach improves on previous methods in both accuracy and processing speed. These results highlight the reliability and practical applicability of our framework for various tasks requiring efficient and accurate QR code processing in dynamic environments.

However, limitations include a restricted scope of verification focusing on 24 types of QR code patterns and a specialization in printed document images. Future research should address these weaknesses by expanding the dataset to encompass a broader range of QR code patterns and exploring techniques for extracting QR codes from digital sources, thus enhancing the framework's versatility and utility in real-world scenarios. To ensure the associated complexity and resource requirements, the potential integration of emerging technologies such as mobile device capabilities and blockchain for real-time processing and enhanced security can be explored for future development and application of the proposed approach.

Journal: **10489** MS: **6509** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2025/3/28** Pages: **22** Layout: **Large**

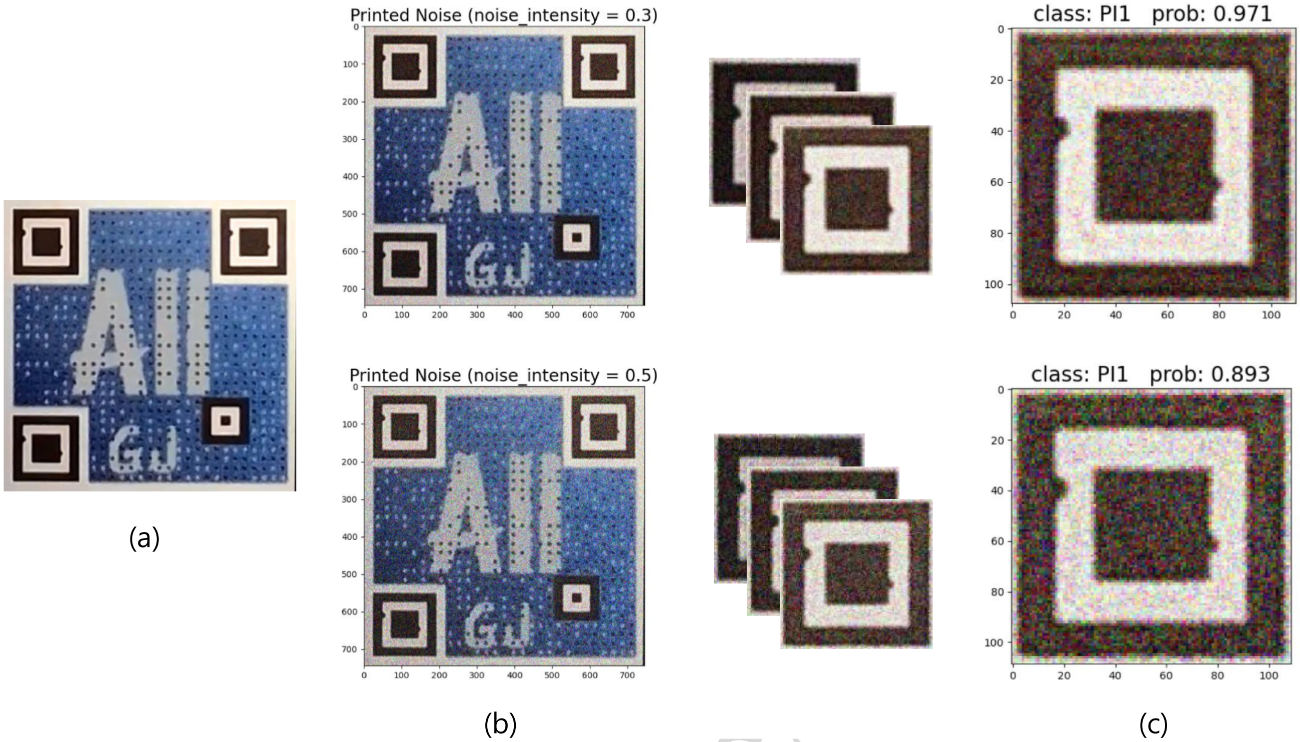


Fig. 15 Model predictions on QR code images with introduced Printed Noise, displaying accuracy rates between 89.3% and 98.4%. This evaluation highlights the proposed QR code authentication model's robustness in handling common real-world scenarios involving printed noise. Despite the introduced noise, the model consistently predicted the correct patterns, demonstrating its effectiveness and reliability for QR code classification

7 Conclusion

In this paper, we presented QR code recognition and verification in challenging imaging conditions, particularly under the influence of different noise. This study introduced a novel two-stage strategy, merging enhanced adaptive thresholding with a cutting-edge deep learning framework, to enable the QR code verification process. Our findings clearly demonstrate the superiority of the proposed methodology over existing approaches, achieving a processing speed of 0.08 seconds and a high accuracy rate of 99.99% in constrained scenarios. Furthermore, the capability of the deep learning model, underpinned by extensive training datasets, to accurately distinguish genuine QR codes from counterfeit versions not only attests to the effectiveness of our methodology but also highlights its potential to reshape the future of QR code authentication in the digital domain.

The robustness of our methodology in varied hardware environments and its energy efficiency have not yet been explored, providing avenues for further investigation. Additionally, as forgery techniques advance, continuous refinement and adaptability of our verification system become imperative. Furthermore, we plan to conduct additional experiments to evaluate system performance across various hardware configurations, ensuring that our approach remains robust and efficient on both high-end and low-end devices. This assessment will help confirm the feasibility of our solution for a wide range of real-world applications. [10]

Acknowledgements This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03038540) and by Institute of Information and communications Technology Planning and Evaluation (IITP) under the metaverse support program to nurture the best talents (IITP-2024-RS-2023-00254529) grant funded by the Korea government(MSIT).

Author Contributions **Nur Alam:** Conceptualization, Investigation, Methodology, Formal analysis, System development, Visualization, Writing - original draft, Writing - review and editing. **A S M Sharifuzzaman Sagar:** Pattern extraction, methodology, Writing - review and editing. **Wenqi Zhang, Taicheng Jin, Arailym Dosset:** training models mobileNet v2, mobileNet v2, and DenseNet. **L. Minh Dang and Moon Hyeonjoon:** Supervised, reviewed, edited.

Availability of Data and Materials The data underlying this article will be shared on reasonable request to the corresponding author.

References

1. Liu H, Zhang C, Deng Y, Xie B, Liu T, Li YF (2023) Transifc: invariant cues-aware feature concentration learning for efficient fine-grained bird image classification. IEEE Trans Multimed
2. Liu H, Liu T, Zhang Z, Sangaiah AK, Yang B, Li Y (2022) Arhpe: Asymmetric relation-aware representation learning for head pose estimation in industrial human-computer interaction. IEEE Trans Industr Inf 18(10):7107–7117
3. Liu H, Fang S, Zhang Z, Li D, Lin K, Wang J (2021) Mfdnet: Collaborative poses perception and matrix fisher distribution for head pose estimation. IEEE Trans Multimedia 24:2449–2460
4. Otsu N et al (1975) A threshold selection method from gray-level histograms. Automatica 11(285–296):23–27
5. Blanger L, Hirata NS (2019) An evaluation of deep learning techniques for qr code detection. In: 2019 IEEE international conference on image processing (ICIP), IEEE, pp 1625–1629
6. Jiang B, Ji Y, Tian X, Wang X (2019) Batch reading densely arranged qr codes. In: IEEE INFOCOM 2019-IEEE conference on computer communications, IEEE, pp 1216–1224
7. He Y, Yang Y (2019) An improved sauvola approach on qr code image binarization. In: 2019 IEEE 11th international conference on advanced infocomm technology (ICAIT), IEEE, pp 6–10 (2019)
8. Zhang J, Min X, Jia J, Zhu Z, Wang J, Zhai G (2021) Fine localization and distortion resistant detection of multi-class barcode in complex environments. Multimed Tools Appl 80:16153–16172
9. Dong H, Liu H, Li M, Ren F, Xie F (2024) An algorithm for the recognition of motion-blurred qr codes based on generative adversarial networks and attention mechanisms. Int J Compu Intell Syst 17(1):1–17
10. Liu H, Nie H, Zhang Z, Li YF (2021) Anisotropic angle distribution learning for head pose estimation and attention understanding in human-computer interaction. Neurocomputing 433:310–322
11. Liu H, Zhang C, Deng Y, Liu T, Zhang Z, Li Y-F (2023) Orientation cues-aware facial relationship representation for head pose estimation via transformer. IEEE Trans Image Process 32:6289–6302
12. Liu T, Yang B, Liu H, Ju J, Tang J, Subramanian S, Zhang Z (2022) Gmdl: Toward precise head pose estimation via gaussian mixed distribution learning for students' attention understanding. Infrared PhysTechnol 122
13. Yan Y, Zou Z, Xie H, Gao Y, Zheng L (2020) An iot-based anti-counterfeiting system using visual features on qr code. IEEE Internet Things J 8(8):6789–6799
14. Ismail S, Alkawaz MH, Kumar AE (2021) Quick response code validation and phishing detection tool. In: 2021 IEEE 11th IEEE symposium on computer applications & industrial electronics (ISCAIE), IEEE, pp 261–266
15. Vinh Loc C, Xuan Viet T, Hoang Viet T, Hoang Thao L, Hoang Viet N (2023) Deep learning based-approach for quick response code verification. Appl Intell 53(19):22700–22714
16. Loc CV, De TC, Burie JC, Ogier JM (2023) An approach for tamper-proof qr code using deep learning based-data hiding. In: Asian conference on intelligent information and database systems, Springer, pp 129–141
17. Hantono BS, Alfarozi SAI, Pratama AR, Rizqi AAA, Mustika IW, Riasetiawan M, Asih AMS (2023) Enhancing counterfeit detection with multi-features on secure 2d grayscale codes. Comput 12(9):183
18. Minh D, Wang HX, Li YF, Nguyen TN (2022) Explainable artificial intelligence: a comprehensive review. Artif Intell Rev 1–66
19. Ohbuchi E, Hanaizumi H, Hock LA (2004) Barcode readers using the camera device in mobile phones. In: 2004 international conference on cyberworlds, IEEE, pp 260–265
20. Hu H, Xu W, Huang Q (2009) A 2d barcode extraction method based on texture direction analysis. In: 2009 5th international conference on image and graphics, IEEE, pp 759–762
21. Dubská M, Herout A, Havel J (2016) Real-time precise detection of regular grids and matrix codes. J Real-Time Image Proc 11:193–200
22. Klimek G, Vamossy Z (2013) Qr code detection using parallel lines. In: 2013 IEEE 14th international symposium on computational intelligence and informatics (CINTI), IEEE, pp 477–481
23. Wu B, Zhong S (2012) A distortion rectification method for distorted qr code images based on self-adapting structural element. In: 2012 8th international conference on wireless communications, networking and mobile computing, IEEE, pp 1–6
24. Huiying D, Tingting H (2015) Transmission line extraction method based on hough transform. In: The 27th chinese control and decision conference (CCDC), IEEE, pp 4892–4895
25. Chou TH, Ho CS, Kuo YF (2015) Qr code detection using convolutional neural networks. In: 2015 international conference on advanced robotics and intelligent systems (ARIS), IEEE, pp 1–5
26. Hou AL, Yuan F, Ying G (2011) Qr code image detection using run-length coding. In: Proceedings of 2011 international conference on computer science and network technology, vol 4. IEEE, pp 2130–2134
27. Ostkamp M, Luzar S, Bauer G (2014) Qr codes on curved media facades: Two approaches for inverse distortion based on raytracing and image warping. In: 2014 international conference on computer graphics theory and applications (GRAPP), IEEE, pp 1–6
28. Ahn M, Hong S, Lee S (2015) A research on the qr code recognition improvement using the cloud-based pre-generated image matching scheme. In: 2015 international conference on information networking (ICOIN), IEEE, pp 356–357
29. Kato Y, Deguchi D, Takahashi T, Ide I, Murase H (2011) Low resolution qr-code recognition by applying super-resolution using the property of qr-codes. In: 2011 international conference on document analysis and recognition, IEEE, pp 992–996
30. Liu Y, Yang J, Liu M (2008) Recognition of qr code with mobile phones. In: 2008 chinese control and decision conference, IEEE, pp 203–206
31. Chu CH, Yang DN, Chen MS (20017) Image stablization for 2d barcode in handheld devices. In: Proceedings of the 15th ACM international conference on multimedia, pp 697–706
32. Chen Q, Du Y, Lin R, Tian Y (2012) Fast qr code image process and detection. In: Internet of things: International workshop, IOT 2012, Changsha, China, August 17-19, 2012. Proceedings, Springer, pp 305–312
33. Belussi L, Hirata N (2011) Fast qr code detection in arbitrarily acquired images. In: 2011 24th SIBGRAPI conference on graphics, patterns and images, IEEE, pp 281–288
34. Xie S, Tan HZ (2021) An anti-counterfeiting architecture for traceability system based on modified two-level quick response codes. Electron 10(3):320
35. Zeggari M, Lambiotte R, Abadi A (2022) An efficient and decentralized blockchain-based commercial alternative (full version). arXiv preprint arXiv:2210.08372
36. Tran DT et al (2015) Rfid anti-counterfeiting for retailing systems. J Appl Math Phys 3(01):1
37. Xu X, Liu Y, Luo D (2022) Optical multiplexing anti-counterfeiting film based on self-assembled and holography lithographic photonic architectures. IEEE Photonics J 14(3):1–5
38. Yiu NC (2016) An nfc-enabled anti-counterfeiting system for wine industry. arXiv preprint arXiv:1601.06372
39. Bala Krishna M, Dugar A (2016) Product authentication using qr codes: a mobile application to combat counterfeiting. Wireless Pers Commun 90:381–398

Journal: **10489** MS: **6509** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2025/3/28** Pages: **22** Layout: **Large**

40. Wan S, Lu Y, Yan X, Liu L (2017) A novel visual secret sharing scheme based on qr codes. Int J Digit Crime Forensic (IJDCF) 9(3):38–48
41. Ma N, Zhang X, Zheng HT, Sun J (2018) Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV), pp 116–131
42. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25
43. Zhang Z, Li J, Shao W, Peng Z, Zhang R, Wang X, Luo P (2019) Differentiable learning-to-group channels via groupable convolutional neural networks. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3542–3551
44. Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6848–6856
45. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258
46. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
47. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
48. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11976–11986
49. Sajol MSI, Hasan J, Islam S, Rahman S (2024) A convnext v2 approach to document image analysis: Enhancing high-accuracy classification. Authorea Preprints
50. Li Y, Hu J, Wen Y, Evangelidis G, Salahi K, Wang Y, Tulyakov S, Ren J (2023) Rethinking vision transformers for mobilenet size and speed. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 16889–16900
51. Gulzar Y (2023) Fruit image classification model based on mobilenetv2 with deep transfer learning technique. Sustain 15(3):1906
52. Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V et al (2019) Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1314–1324
53. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
54. Xu J, Pan Y, Pan X, Hoi S, Yi Z, Xu Z (2022) Regnet: self-regulated network for image classification. IEEE Trans Neural Netw Learn Syst
55. Mao X, Qi G, Chen Y, Li X, Duan R, Ye S, He Y, Xue H (2022) Towards robust vision transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 12042–12051
56. Koonce B, Koonce B (2021) Efficientnet. Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization, pp 109–123
57. Yang J, Yang G (2018) Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer. Algorithm 11(3):28
58. Wang H, Li Y, Dang LM, Ko J, Han D, Moon H (2020) Smartphone-based bulky waste classification using convolutional neural networks. Multimed Tools Appl 79:29411–29431
59. Vinh Loc C, Xuan Viet T, Hoang Viet T, Hoang Thao L, Hoang Viet N (2023) Deep learning based-approach for quick response code verification. Appl Intell 53(19):22700–22714
60. Sambare S, Pathak U, Sonavane V, Gadiya M, Musale S (2023) Document verification system and validation against qr codes. In: 2023 7th international conference on computing, communication, control and automation (ICCUBEA), IEEE, pp 1–5
61. Tsai MJ, Chen TM (2021) A deep learning approach for qr code based printed source identification. In: 2021 15th international conference on signal processing and communication systems (ICSPCS), IEEE, pp 1–6

Journal: **10489** MS: **6509** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2025/3/28** Pages: **22** Layout: **Large**

Journal: 10489
Article: 6509

Author Query Form

Please ensure you fill out your response to the queries raised below and return this form along with your corrections

Dear Author

During the process of typesetting your article, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query	Details required	Author's response
1.	Please check if the captured title is correct.	Correct
2.	Please check and confirm if the author names and initials are correct	Correct
3.	Please check and confirm if the authors and their respective affiliations have been correctly identified. Amend if necessary.	Checked and Confirmed
4.	Photography and Biography are desired. Please provide. Otherwise, please confirm if unnecessary.	Not Provided
5.	Missing citation for Figures 1 was inserted here. Please check if appropriate. Otherwise, please provide citation for Figures 1. Note that the order of main citations of figures in the text must be sequential.	Correct
6.	A citation for Fig. 2 and 3 was inserted here so that figure citations are in sequential order. Please check if appropriate. Otherwise, insert citation(s) in sequential order.	Correct
7.	Please check all the Table and their Table data if captured and presented correctly.	Checked
8.	Missing citation for Figure 5 was inserted here. Please check if appropriate. Otherwise, please provide citation for Figure 5. Note that the order of main citations of figures in the text must be sequential.	It should be at the end of line 293
9.	Please check all inline and displayed equations if captured and presented correctly.	Checked
10.	Inclusion of a "Competing Interests" statement is preferred for this journal. If applicable, please provide the statement.	Please add this "The authors declare no competing interests."