symmetry

Article

An Efficient Machine Learning-Based Model to Effectively Classify the Type of Noises in QR Code: A Hybrid Approach

Jawad Rasheed [1,*], Ahmad B. Wardak [2], Adnan M. Abu-Mahfouz [3,4], Tariq Umer [5], Mirsat Yesiltepe [6] and Sadaf Waziry [2]

1 Department of Software Engineering, Nisantasi University, 34398 Istanbul, Turkey
2 Department of Computer Engineering, Istanbul Aydin University, 34295 Istanbul, Turkey
3 Council for Scientific and Industrial Research (CSIR), Pretoria 0184, South Africa
4 Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2006, South Africa
5 Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore 54000, Pakistan
6 Department of Mathematical Engineering, Yildiz Technical University, 34220 Istanbul, Turkey
* Correspondence: jawad.rasheed@nisantasi.edu.tr

Abstract: Granting smart device consumers with information, simply and quickly, is what drives quick response (QR) codes and mobile marketing to go hand in hand. It boosts marketing campaigns and objectives and allows one to approach, engage, influence, and transform a wider target audience by connecting from offline to online platforms. However, restricted printing technology and flexibility in surfaces introduce noise while printing QR code images. Moreover, noise is often unavoidable during the gathering and transmission of digital images. Therefore, this paper proposed an automatic and accurate noise detector to identify the type of noise present in QR code images. For this, the paper first generates a new dataset comprising 10,000 original QR code images of varying sizes and later introduces several noises, including salt and pepper, pepper, speckle, Poisson, salt, local var, and Gaussian to form a dataset of 80,000 images. We perform extensive experiments by reshaping the generated images to uniform size for exploiting Convolutional Neural Network (CNN), Support Vector Machine (SVM), and Logistic Regression (LG) to classify the original and noisy images. Later, the analysis is further widened by incorporating histogram density analysis to trace and target highly important features by transforming images of varying sizes to obtain 256 features, followed by SVM, LG, and Artificial Neural Network (ANN) to identify the noise type. Moreover, to understand the impact of symmetry of noises in QR code images, we trained the models with combinations of 3-, 5-, and 7-noise types and analyzed the classification performance. From comparative analyses, it is noted that the Gaussian and Localvar noises possess symmetrical characteristics, as all the classifiers did not perform well to segregate these two noises. The results prove that histogram analysis significantly improves classification accuracy with all exploited models, especially when combined with SVM, it achieved maximum accuracy for 4- and 6-class classification problems.

Keywords: quick response; noisy images; noise classification; CNN; histogram analysis; SVM

Citation: Rasheed, J.; Wardak, A.B.; Abu-Mahfouz, A.M.; Umer, T.; Yesiltepe, M.; Waziry, S. An Efficient Machine Learning-Based Model to Effectively Classify the Type of Noises in QR Code: A Hybrid Approach. *Symmetry* **2022**, *14*, 2098. https://doi.org/10.3390/sym14102098

Academic Editor: Dumitru Baleanu

Received: 15 August 2022
Accepted: 5 October 2022
Published: 8 October 2022

1. Introduction

The quick response (QR) code is a form of two-dimensional barcode that was initially developed in Japan for the automotive industry [1]. Denso Wave, a Japanese manufacturer, invented the matrix barcode in 1994 [2]. It has the advantages of a vast amount of information, high reliability, a diverse variety of information, such as text and images, and good security [3]. Vendors are becoming more interested in QR codes as they emerge [4]. However, the QR code's original appearance was not meant for human perception.

People cannot read barcodes with their eyes since a standard QR code only contains black and white modules. The construction of the barcode makes it impossible to change its appearance. By scanning the coding, we may instantly get specific information. However, noise in the printed image is unavoidable owing to printer processes and restricted printing technology. Noise is often introduced during the gathering and transmission of digital images [5]. Various noises, such as Poisson, Salt and Pepper, Gaussian, Speckle noise, and others, may decrease the sharpness of a QR code image. These noises are caused by improper memory allocation, compression, a short focal length, post-filtering, and other undesirable environmental or image-capturing equipment conditions.

However, there is a need for some efficient methods to correctly identify various noise kinds so that they may be easily eradicated. In this work, we will be using machine learning, deep learning techniques, such as Convolutional Neural Networks (CNNs), Artificial Neural Networks (ANNs), Support Vector Machine (SVM), and Logistic Regression (LG) for the classification of QR code noises. Machine Learning is now a popular issue in the technology sector [6] and for good reason: it represents a huge leap in the way computers learn. Machine Learning is becoming more popular as technology advances and vast amounts of data, called Big Data, become available. Many recent image processing approaches employ Machine Learning models, such as Deep Neural Networks, to modify images for a variety of purposes, such as adding creative filters, optimizing an image for quality, or refining certain image aspects for computer vision applications [7].

Deep learning (DL) is an area of machine learning [8] that has recently become one of the most significant achievements and research hotspots. CNNs, a form of deep learning neural network, offer a substantial improvement in image identification. To perform its function, a CNN pulls features from images. This removes the need for manual feature extraction. As the network trains on a sequence of images, the features are acquired. CNNs learn to recognize features by switching between tens or hundreds of hidden layers. This is a multi-layer neural network composed of neurons with trainable weights and biases [9] and it is made possible by powerful GPUs that enable us to stack deep layers and handle a wide range of image input properties [10]. LG has been widely used as a complete data processing strategy for binary classification and prediction [11]. Logistic regression is mostly used to categorize data and its data points are not structured in rows. In comparison, SVM is a well-known pattern recognition and image classification approach [12]. Based on a kernel function, it produces the most effective separating hyperplanes. Each data item in the SVM technique is plotted as a point in n-dimensional space, where n is the number of features and the value of each feature is the value of a certain coordinate. Various other works about noise removal have been conducted in a medical domain, such as [13–14].

Noisy images are hazardous to the training of neural networks and other techniques, decreasing the classification performance of the networks [10]. Image noise may be either cumulative or progressive [15]. The cumulative noise model adds a noise signal to the original signal to produce a severely corrupted noisier signal, while the progressive noise model multiplies the original signal by the noise signal. Therefore, this study investigated advanced deep learning models, such as CNN and classical machine learning classifiers (SVM and LG), to classify the noises present in QR code images, first by uniformly resizing images of varying sizes. In addition to this, the study proposed an amalgam approach comprising histogram density analysis and machine learning classifiers (ANN, SVM, and LG) that works well even for images with varying sizes. Such noise identification systems can later help researchers and programmers to apply specific noise removal filters concerning noise identified, so that original data/information can be retrieved from available QR codes. Likewise, application developers in smart product manufacturing industries can enhance their scanners by supporting features of retrieving information from a noisy QR code. Following are the major contributions of this study:

- Generates a new dataset containing images of QR codes;

- Enhances the dataset by introducing seven various noises (Speckle, Localvar, Salt, Pepper, Gaussian, Poisson, Salt, and Pepper) to each original QR code image;
- Presents detailed structure for embedding noises in QR code images;
- Analyses the performance of classifiers (CNN, SVM, and LG), trained over images with a fixed size, to distinguish original QR code image from noisy image and identify the noise present, if any;
- Incorporates histogram analysis to transform the features, trace and extract the most relevant features accordingly;
- Widens the investigation by proposing an amalgam approach based on histogram analysis with common machine learning classifiers (ANN, SVM, and LG) to handle images of varying sizes;
- Outlines detailed analysis of proposed scheme performance for three scenarios, when trained with combinations of 3-, 5-, and 7-noise types of images along with original QR code images.

The paper is further structured as follows: In Section 2, the related works of this issue are discussed. Section 3 discusses the proposed methodology. Section 4 covers the experiment results and the comparison/discussion. The conclusion is described in Section 5.

2. Related Work

Various studies have attempted to categorize noise in images to the best of our knowledge, for instance, VGG-16 and Inception-v3 convolutional neural networks were used in the study of [16] to automatically identify noise distributions and it was discovered that Inception-v3 effectively detects the noise distribution out of nine possible distributions: salt and pepper, Gaussian, speckle, exponential, lognormal, uniform, Erlang, Rayleigh, and Poisson. The performance of FFDNet was then compared to that of the noise clinic for each of the noisy image sets. They observed that CNN-based denoising is superior to blind denoising in general, with a 16% improvement in peak signal-to-noise ratio on average (PSNR). In [17], the authors present a noise-robust CNN (NR-CNN) for classifying noisy images without any pre-processing for noise reduction and to improve convolutional neural network classification performance for noisy images. Experiment results reveal that the proposed CNN outperforms VGG-Net-Medium, VGG-Net-Slow, GoogleNet, and ResNet in the classification of noisy pictures. Furthermore, the proposed CNN does not need any pre-processing for noise reduction, which speeds up the classification of noisy images.

Authors in [18] investigated a DNN-based noisy image classification approach, in which five supervised deep learning architectures were utilized to classify the reconstructed picture: DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN, DAE-CDAE-CNN, and DVAE-CDAE—CNN. It was revealed that the first three algorithms perform well on images with low noise levels, but the latter two approaches perform well on enormous volumes of noisy data. The authors of [19] showed how to distinguish different types and intensities of visual noise using a (CNN) technique, as well as a backpropagation algorithm and stochastic gradient descent optimization methodologies. The principal component analysis (PCA) filters generation technique is used to collect data-adaptive filter banks to lower the algorithm's training time and processing cost. Researchers in [20] evaluated image quality assessment (IQA) techniques, fitting curves, mean opinion score (MOS), and the development of two neural networks to provide an Image Noise Level Classification (INLC) strategy for diverse application situations. They explored the reasons for the low classification accuracy and suggested a mild approach of creating a tolerance rate to get higher acceptable accuracy. Milan Tripathi [21] created, implemented, and evaluated a CNN-based classifier to detect noisy images with high validation and training accuracy, as well as a UNET-based model to denoise images with ideal PSNR and SSIM values.

By scanning the QR code, we can receive accurate information in real time. The typical QR code, which is composed of black and white modules, is unsightly and difficult to see. In recent years, there has been an increase in the usage of graphic QR codes in product packaging and marketing activities. When the user scans the printed visual QR code, it is accompanied by a noise phenomenon that interferes with identification and causes failure. As a result, we are working on building an intelligent image noise-type identification technique, as no such system exists in the literature. The reasoning for this is that after the type of noise infecting an image has been properly defined, an appropriate noise-reduction filter can be applied. Although the capacity to minimize noise is crucial, it is equally necessary to identify the type and amount of noise present in QR code images. To address this issue, we proposed (CNN, ANN, SVM, LG)-based models to effectively classify the type of noises (Gaussian, Localvar, Pepper, Poisson, Salt, Speckle, and Salt and Pepper) in QR code images and also extracted the features of images manually using histogram density feature extraction and fed the data to the mentioned models.

3. Proposed Methods

The purpose of this paper is to develop classification models that accept various forms of QR code images as input and categorize them as original QR codes or noisy by anticipating the type of noise. Figure 1 depicts the overall process of the proposed experiment. Because no such dataset exists in the literature, this study created its own QR code image dataset and added seven distinct noises (Gaussian, localvar, pepper, Poisson, speckle, salt, salt & pepper) to the generated original images. The suggested deep learning-based CNN architecture, an ANN model, SVM, and Logistic Regression algorithms are then trained to identify noisy images by accurately predicting their category. The study intends to analyze produced QR code images, scale images, and map noise to the original QR code images (Figure 2), encode labels, build histogram density features of images, train the suggested four distinct models, recognize the type of noise, and output the classified category. We utilized the deep learning framework TensorFlow [22] to tackle the noise-type classification task.

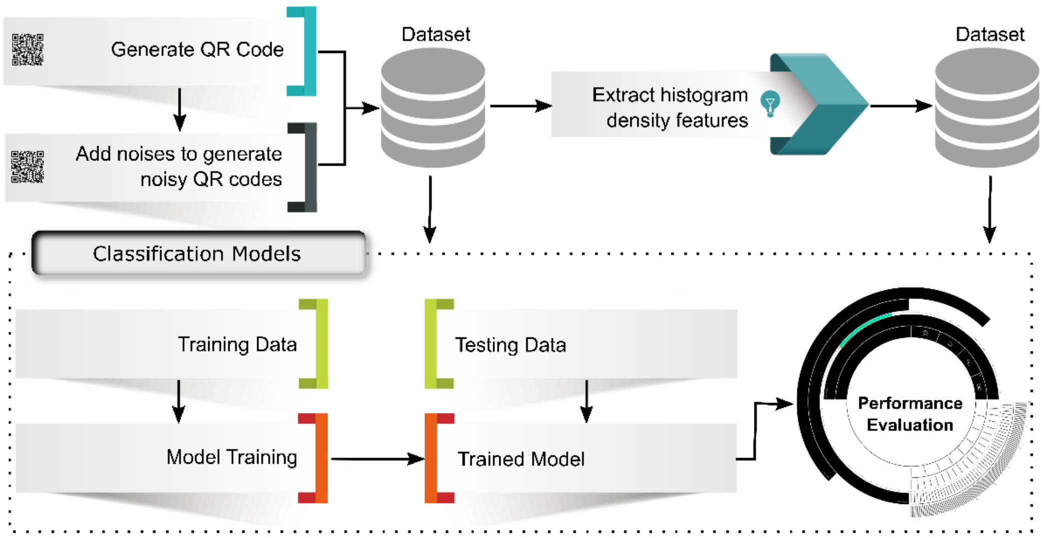


Figure 1. The workflow of the proposed study.

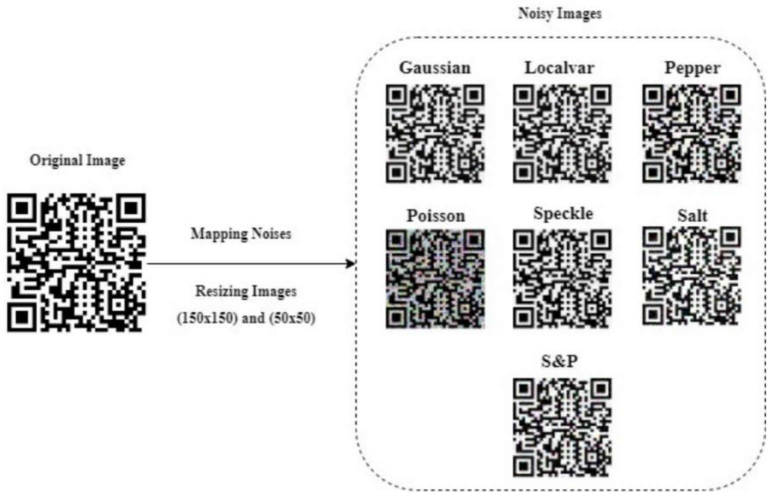


Figure 2. The original quick response code images with varying sizes are resized to 150 × 150 and 7 types of noises are mapped, each separately as one class.

3.1. Data Analyzation

The dataset we are utilizing for this work contains 80,000 images of both original and noisy QR codes. The collection is quite variable; certain classes contain images of varying sizes and quantities and the entire dataset is around 16 GB in size. The dataset's images are in bitmap format (BMP). Because our CNN model requires a fixed-size input, we preprocessed the images to a fixed size of (150 × 150) ratio and (50 × 50) ratio for the SVM and Logistic Regression models. Figure 2 depicts the process of scaling images and mapping different forms of noise to the original images.

We generated normal/original QR code images with random data and then introduced 7 different noises to expand the dataset. Our dataset is divided into 8 distinct classes. The original one and 7 other classes which we generated by adding 7 different types of noises such as Gaussian, Localvar (white noise with a zero-mean Gaussian distribution and an intensity-dependent variance), Salt and Pepper, Poisson, Pepper, Salt, and Speckle to the original type. The following subsections briefly outline the details about each noise type we exploited for this paper.

3.1.1. Gaussian Noise

Gaussian noise [23] is a kind of statistical noise with a probability density function equal to the standard deviation, often known as the Gaussian distribution; in other words, the possible values of the noise are Gaussian distributed. It is named after Carl Friedrich Gauss. A Gaussian distribution's probability density function has a bell-shaped curve. Gaussian noise is most often used in additive white Gaussian noise. The probability density function ϱ of a Gaussian random variable g is given in Equation (1): where g denotes the grey level, μ the mean grey value, and σ the standard deviation. Figure 3 depicts the Gaussian probability distribution function of Gaussian noise and its pixel representation.

$$\rho_{G(g)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(g-\mu)^2}{2\sigma^2}} \quad (1)$$

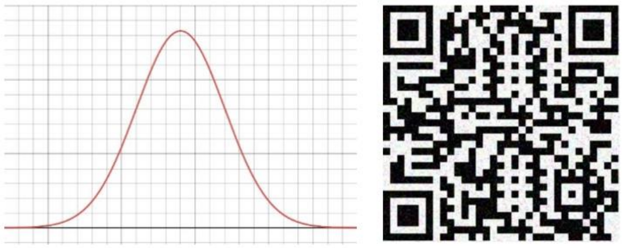


Figure 3. Gaussian noise probability distribution function and its quick response code image sample.

3.1.2. Salt and Pepper Noise

The phrase "salt and pepper noise" refers to a wide range of procedures that all result in the same fundamental image deterioration [24]. It is sometimes referred to as impulse noise. Sharp and rapid disruptions in the visual signal might create this noise. It looks like white and black pixels that are poorly dispersed. The salt and pepper noise has two values, a and b. The values a and b in salt and pepper noise are not the same. Each one has a chance of less than 0.1 on average. The damaged pixels shift between the lowest and highest value, giving the picture a "salt and pepper" appearance. Figure 4 depicts the Salt and Pepper noise and its Probability Distribution Function having a deviation of 0.05 [25].

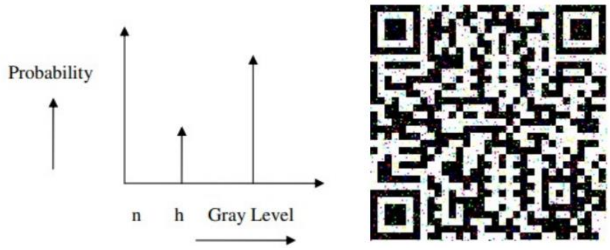


Figure 4. Salt and pepper probability distribution function and its quick response code image sample.

3.1.3. Speckle Noise

Speckle noise is a kind of additive noise, as opposed to Gaussian or salt and pepper noise [25]. This decreases image quality in diagnostic testing by causing images to have a backscattered wave look, created by multiple little dispersed reflections traveling through inner organs. Consequently, the observer's ability to discern minute features in the images is impaired. Speckle noise has a gamma distribution function and is expressed mathematically, as depicted in Equation (2) [25].

$$s(g) = \frac{\left[g^{(\alpha-1)} \times e^{-g/a}\right]}{\{(\alpha - 1)!\, a^{\alpha}\}} \tag{2}$$

where α is the variance and g is the gray level measurement. Figure 5 depicts the gamma distribution function and pixel representation of speckle noise.

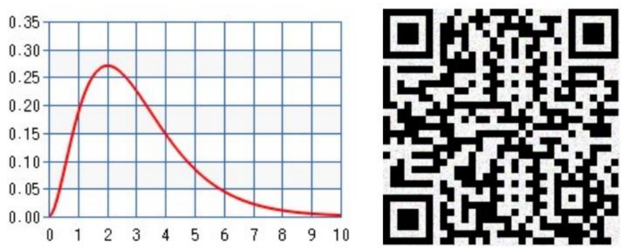


Figure 5. Speckle Gamma probability distribution function and its quick response code image sample.

3.1.4. Poisson Noise

Poisson noise, also known as Shot noise, is a kind of noise that may be represented mathematically using the Poisson process [26]. The nonlinear reactions of image detectors and recorders generate Poisson noise and the image data determine this kind of noise. The discrete nature of electric charge causes shot noise in electronics. Shot noise may also be detected in photon enumeration in optical systems, which is related to light's particle nature. This sort of noise is sometimes referred to as quantum (photon) noise. Poisson noise has a Poisson distribution function, which is a probability distribution used to indicate the frequency with which an event is expected to occur over a certain period as given in Equation (3), where e is Euler's number (2.71828), n represents the number of occurrences, $n!$ is the factorial of n, and λ is equal to n 's anticipated value when it is also equal to its variance. Figure 6 depicts the Poisson noise and its probability distribution function having a deviation of 0.03 in 100 random trials.

$$p(n) = \frac{\lambda^n}{n!} e^{-\lambda} \tag{3}$$

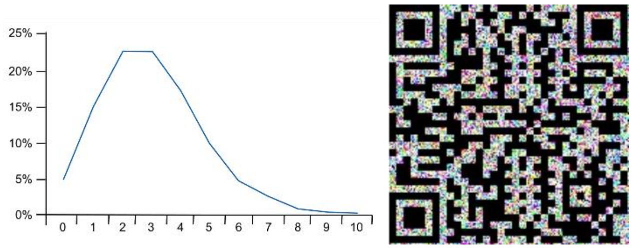


Figure 6. Poisson probability distribution function and its quick response code image sample.

Table 1 lists the division of samples in each class of formed dataset.

Table 1. Quick response code image dataset information.

Label/Class	Number of Samples
Normal/Original QR code	10,000
Gaussian	10,000
Localvar	10,000
Pepper	10,000
Poisson	10,000
Speckle	10,000
Salt	10,000
Salt and pepper	10,000
Total	**80,000**

3.2. Classification Algorithms

3.2.1. Convolutional Neural Network

To categorize the noises in QR code images, we developed a CNN model. The created network hyperparameter tuning is shown in Table 2. Each Conv2D and MaxPooling2D layer produces a three-dimensional (3D) form tensor (height, width, channels). The width and height measurements decrease as we move further into the network. The first argument specifies how many output channels each Conv2D layer has. The max-pooling layer is usually utilized to reduce the output volume's spatial dimensions. In general, as the width and height decrease, we can add more output channels to each Conv2D layer. The Dropout layer helps to reduce overfitting by randomly changing input units to 0 at a frequency of rate throughout the training period. The SoftMax layer normalizes the preceding layer's output by including the likelihood of the actual input picture belonging to recognized classes.

Table 2. Proposed convolutional neural network and its hypermetric values.

Hyperparameter	Value
Optimizer	Adam
Number of epochs	100
Batch size	120
Loss	Categorical cross-entropy
Metrics	Accuracy
Learning rate	0.000001

After extensive experimental combinations, the best CNN model is composed of 9 layers, starting the layers with 32 filters, then 64, and 128 filters, with 3 × 3 kernels. It has a max-pool layer after each convolutional layer and a flatten layer to change the dimensional array for inputting to the dense layer, followed by a dropout layer to get rid of overfitting and then the SoftMax (output) layer with 8 outputs.

3.2.2. Artificial Neural Network

To categorize the noises in histogram feature-extracted QR code images, we developed an ANN model. The hyperparameter tuning of our ANN model is the same as the proposed CNN model (see Table 1). As each feature extracted image contains 256 features; therefore, instead of an input shape of an exact size, which we added for the CNN model, we are adding an input dimension of 256 in the first layer of the model.

To achieve the best performance, we repeatedly performed numerous experiments and found that ANN having 5 layers with 2048, 2024, 128, 128, and 8 units/filters, respectively, attain high accuracy.

3.2.3. Logistic Regression

Before feeding data to the LG model, we scaled and rearranged our data, then used the Standard Scaler to resize the distribution of values such that the mean of the observed values is 0 and the standard deviation is 1. Thus, it eliminates the mean and scales each feature to unit variance. The designed LG network architecture is presented in Table 3. For each candidate, the training is performed over 10-fold cross-validation having a total of 10 fits with a parallel (−1) number of tasks utilizing backend (LokyBackend) with 8 concurrent workers.

Table 3. Proposed logistic regression model and its hypermetric values.

Hyperparameter	Value
CV	10
No. of jobs	−1
Random state	1234
Max. iteration	1000 and 400 (feature extracted data)
Solver	Liblinear
Class weight	Balanced
Verbose	1

3.2.4. Support Vector Machine

We pre-processed data in the SVM [27] model, such as scaling it to a 50 × 50 ratio, and then used the Standard Scaler to resize the distribution of values such that the mean of the observed values is 0 and the standard deviation is 1 and shuffled the data to reorder the order of the items. The developed SVM network architecture is shown in Table 4.

Table 4. Proposed support vector machine model and its hypermetric values.

Hyperparameter	Value
C	1
Gamma	Auto
Kernel	Poly and RBF (feature extracted data)

The error term's penalty parameter, C, is set to 1 to manage the error, while Gamma is set auto to supply the decision boundary curvature weight. Moreover, the poly kernel is used to describe the similarity of training samples in feature sets across polynomials for the original variables, allowing for the learning of nonlinear models.

3.3. Histogram Density Feature Extraction

The histogram density values in the grayscale state of the images are employed as features in this technique. In this scenario, the ratio of the number of pixels with each grey tone to the total number is utilized as a feature value, such that each image contains 256 features as depicted in Figure 7. After extracting the features for each image, we save it as a new dataset and then feed it to the proposed ANN, SVM, and LG models to classify the noise types of QR codes.

0	1	2	3	4	5	6	7	8	9	...	252	253	254	255
0.509236	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.000000	0.490764
0.502613	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.000000	0.497387
0.509525	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.000000	0.490475
0.491728	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.000000	0.508272
0.509075	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.000000	0.490925
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0.495140	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.025987	0.034252	0.039256	0.102427
0.502155	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.024783	0.033634	0.038437	0.103509
0.525101	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.024139	0.032428	0.037440	0.095332
0.501453	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.024503	0.034898	0.038772	0.102789
0.507694	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.025553	0.034038	0.039034	0.099178

Figure 7. Sample of histogram density features of quick response code images.

4. Results and Discussions

To evaluate the model's accuracy, we used the dataset that contains 80,000 images of QR codes with various types of noises. The dataset was divided into the train and test sets with ratios of 70/30, respectively.

Before proposing the suggested scheme, we tested several state-of-the-art pre-trained deep learning-based models to segregate QR code images into normal and seven noisy QR code images. Table 5 shows performance of these models on the generated dataset.

Table 5. Performance of various state-of-the-art deep learning models.

Model/Network	Accuracy
VGG16	81.77
ResNet18	85.24
SqueezeNet	81.26
MobileNetV2	84.52
DenseNet121	85.75

As depicted in Figure 1, the paper proposed two different schemes for the classification of noisy QR images. In the first scheme, the generated dataset images (original QR code, and noisy QR code) are fed directly to various deep learning and machine learning-based classification models. The second scheme exploited the histogram density feature extraction technique to shape data into useful representation and then fed it to classification models. After successful training, the accuracy of all models in proposed schemes is computed using all data from the test dataset. Our model's usefulness and performance are evaluated using four metrics: accuracy, precision, recall, and f1 score. Moreover, confusion matrices are also presented to analyze the false-positive and false-negative test data. The experimental study used cross entropy as a loss function, specifically the categorical type of cross entropy because a stable loss function will generalize the model well [28].

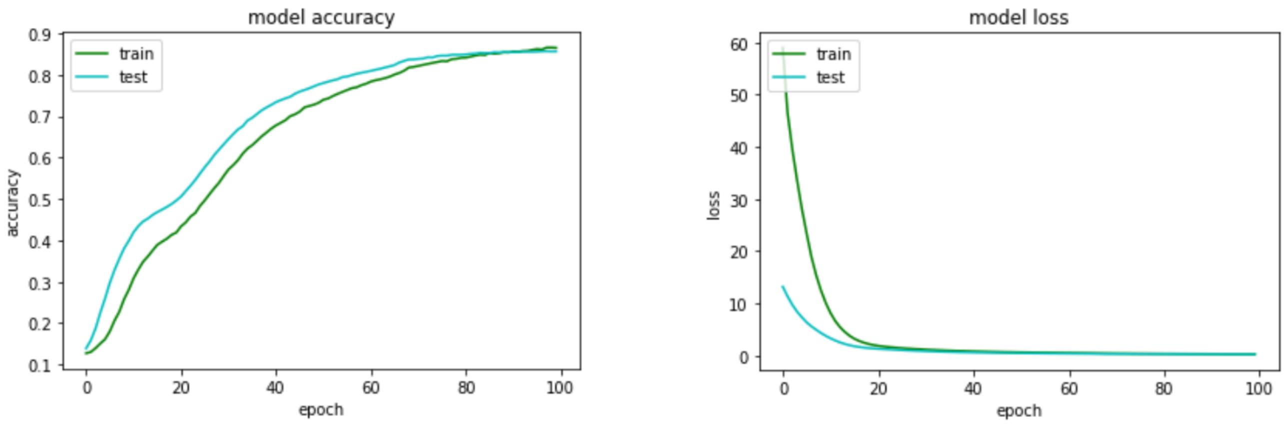
4.1. Scheme 1: Classification Using Deep and Machine Learning Models without Feature Extraction

This scheme does not involve any computer vision technique to extract useful features and the generated QR code images are directly fed to machine learning and deep learning classification models, including CNN, SVM, and LG. We further widened the study by introducing three different scenarios of the dataset, where each scenario involves a combination of different types of noisy images with original QR images to train the classification model.

4.1.1. Scenario 1 (8-Class Classification Task)

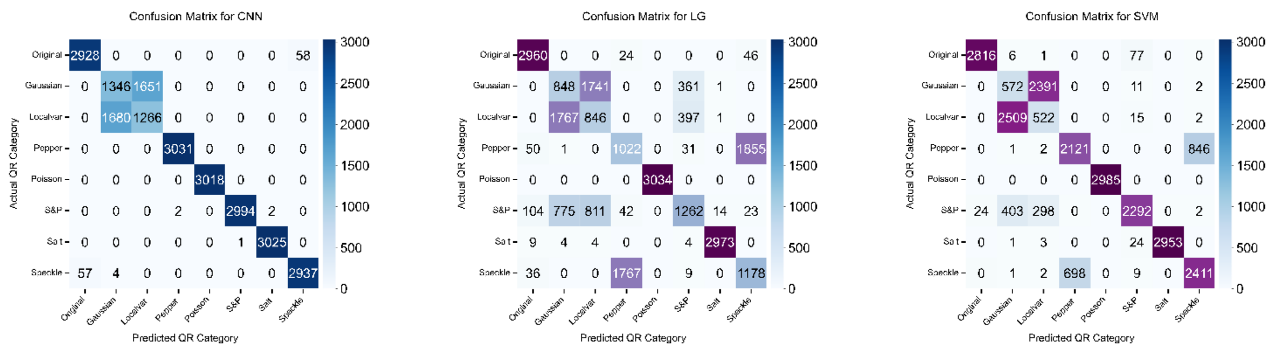
In this scenario, we utilized all eight types of QR code images, the original QR code, and mapped seven noises (Gaussian, Localvar, Pepper, Poisson, Speckle, Salt, Salt and pepper) to the original images. The images are fed to three classification models separately. Figure 8 depicts the performance curves of the proposed CNN.

Figure 9 depicts the confusion matrices for CNN, LG, and SVM, whereas Table 6 presents the overall performance of trained models. It is evident that CNN performed better than LG and SVM by obtaining an overall accuracy of 85.6%; however, confusion matrices show that models' performances degraded as they are unable to distinguish between Gaussian and localvar noisy images. Evidently, the obtained matrices for analyzing the effectiveness and performance of the trained models for eight types of QR code images show that such a proposed scheme is not suitable for this scenario as it hardly segregates images containing localvar and Gaussian noises.



(**a**) (**b**)

Figure 8. Performance curves of convolutional neural network model when trained without external feature extraction technique to classify 8 types of quick response code images: (**a**) accuracy curves for train and test sets; (**b**) loss curves for train and test sets.



(**a**) (**b**) (**c**)

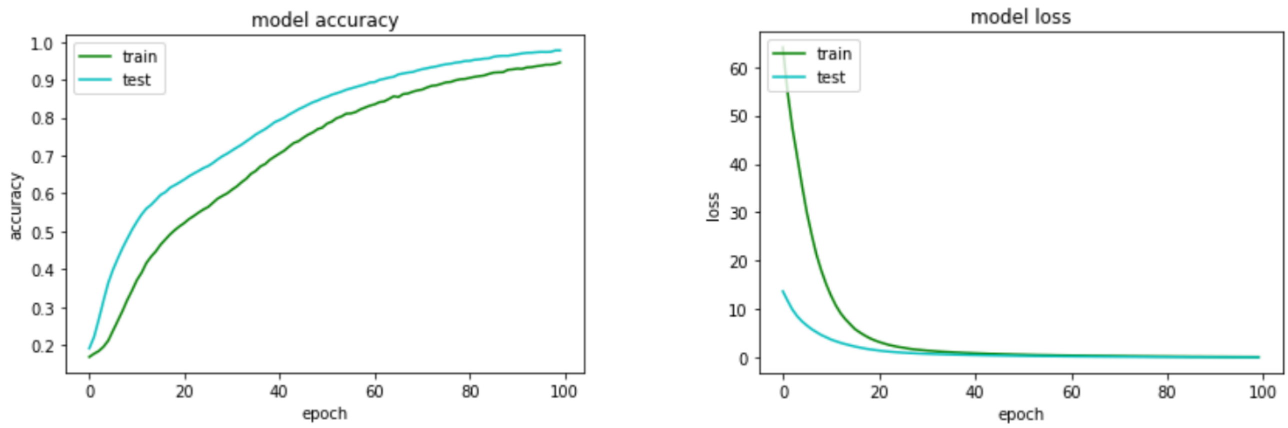
Figure 9. The confusion matrices for proposed classifiers when trained without external feature extraction technique to classify original quick response (QR) code and 7 different types of noisy QR code images. (**a**) convolutional neural network (CNN); (**b**) logistic regression (LG); and (**c**) support vector machine (SVM).

Table 6. Performance comparison between convolutional neural network (CNN), logistic regression (LG), and support vector machine (SVM) when trained without external feature extraction technique to classify 8 types of quick response code images.

Model	Accuracy	Precision	Recall	F1-score
CNN	**85.60%**	85.50%	85.50%	85.60%
LG	58.85%	59.74%	58.67%	58.94%
SVM	69.74%	71.91%	69.64%	70.62%

4.1.2. Scenario 2 (6-Class Classification Task)

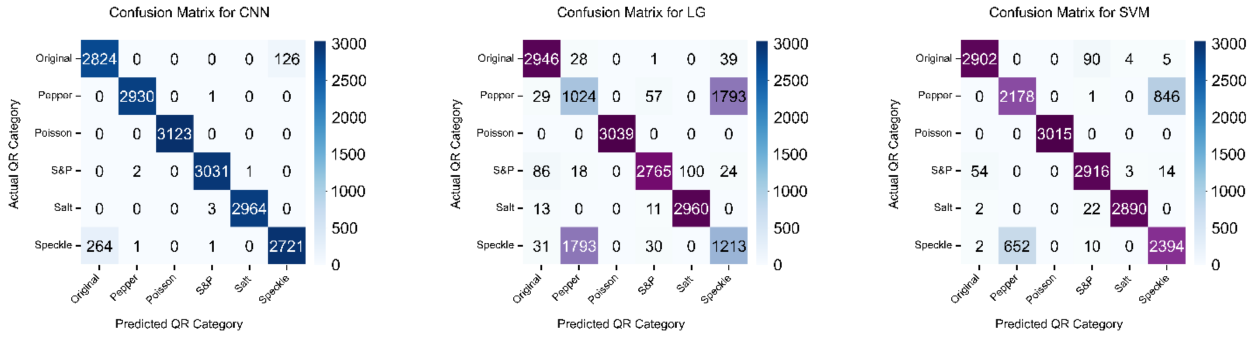
In this scenario, we removed two types of noisy images (Gaussian and localvar) and used the other six types of QR code images, the original QR code and noisy types of QR images (pepper, Poisson, speckle, salt, and salt and pepper). Similarly, the images are fed to three classification models separately. All models are trained and tested over 70% and 30% of data, respectively. Figure 10 depicts the performance curves of the proposed CNN model.



(**a**) (**b**)

Figure 10. Performance curves of convolutional neural network model when trained without external feature extraction technique to classify 6 types of quick response code images: (**a**) accuracy curves for train and test sets; (**b**) loss curves for train and test sets.

Figure 11 shows the confusion matrices for CNN, LG, and SVM. Table 7 presents the overall performance of trained models. It is evident that the exploited models performed better for six types of QR code image classification tasks. All the models gained accuracy as compared to Scenario 1. Again, CNN performed better by obtaining an overall accuracy of 97.74%, whereas SVM reached 90.53% accuracy while LG hardly achieved an overall accuracy of 77.48%. Moreover, the confusion matrices show that models did not perform well while segregating pepper and Speckle noisy images.



(**a**) (**b**) (**c**)

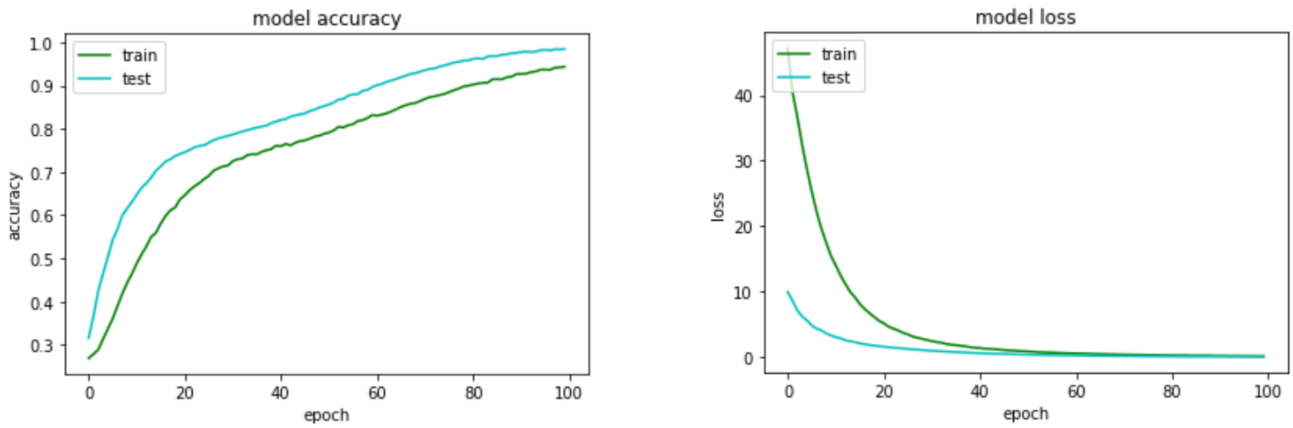
Figure 11. The confusion matrices for proposed classifiers when trained without external feature extraction technique to classify original quick response (QR) code and 5 different types of noisy QR code images. (**a**) convolutional neural network (CNN); (**b**) logistic regression (LG); and (**c**) support vector machine (SVM).

Table 7. Performance comparison between convolutional neural network (CNN), logistic regression (LG), and support vector machine (SVM) when trained without external feature extraction technique to classify 6 types of quick response code images.

Model	Accuracy	Precision	Recall	F1-score
CNN	**97.70%**	97.70%	97.70%	97.70%
LG	77.48%	77.24%	77.36%	77.29%
SVM	90.53%	90.70%	90.63%	90.67%

4.1.3. Scenario 3 (4-Class Classification Task)

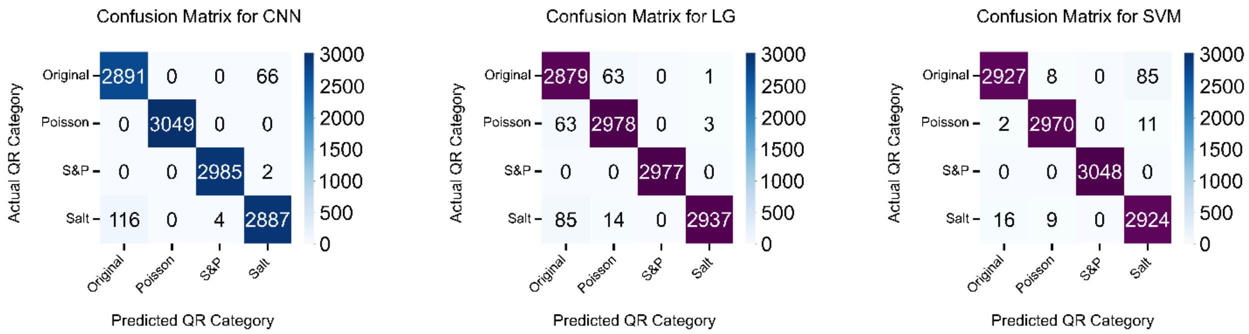
In this scenario, we removed four types of noisy images (Gaussian, localvar, pepper, and speckle) and used the rest of the four types of QR code images, the original QR code and three noisy types of QR images (Poisson, salt, and salt and pepper). The images are fed to three classification models separately. All models are trained and tested over 70% and 30% of data, respectively. Figure 12 depicts the performance curves of the proposed CNN model.



(**a**) (**b**)

Figure 12. Performance curves of convolutional neural network model when trained without external feature extraction technique to classify 5 types of quick response code images: (**a**) accuracy curves for train and test sets; (**b**) loss curves for train and test sets.

Figure 13 shows the confusion matrices for CNN, LG, and SVM. Table 8 presents the overall performance of trained models. The exploited classification models performed better for four types of QR code image classification tasks. All the models gained higher accuracy as compared to Scenario 1 and Scenario 2. While classifying these four types of QR codes (original, Poisson, salt, and salt and pepper), all models attained almost similar performance; however, precisely, SVM topped by obtaining an overall accuracy of 98.91%, whereas CNN reached 98.43% accuracy while LG competes by accomplishing an overall accuracy of 98.09%. Moreover, the confusion matrices show that models are suitable for datasets composed of the four types of QR code images.



(**a**) (**b**) (**c**)

Figure 13. The confusion matrices for proposed classifiers when trained without external feature extraction technique to classify original quick response (QR) code and 3 different types of noisy QR code images. (**a**) convolutional neural network (CNN); (**b**) logistic regression (LG); and (**c**) support vector machine (SVM).

Table 8. Performance comparison between convolutional neural network (CNN), logistic regression (LG), and support vector machine (SVM) when trained without external feature extraction technique to classify four types of quick response code images.

Model	Accuracy	Precision	Recall	F1-score
CNN	98.43%	98.40%	98.40%	98.40%
LG	98.09%	98.11%	98.10%	98.90%
SVM	**98.91%**	98.91%	98.91%	98.90%

4.2. Scheme 2: Classification Using Deep and Machine Learning Models with Feature Extraction

This scheme involves a computer vision technique to extract useful features. It exploited the histogram density analysis technique to extract useful features from each dataset image. Figure 14 shows the histogram analysis for all kinds of QR code images in the dataset, where it extracted 256 features using histogram density feature extraction, which is then fed to machine learning-based classification models, including ANN, SVM, and LG. Contrary to Scheme 1, the dataset is no longer in the form of images; thus, ANN is exploited instead of CNN under this scheme, such as Scheme 1. We further widened the study by introducing three different scenarios of the dataset, where each scenario involves a combination of different types of noisy images with original QR images to train and test the classification model.

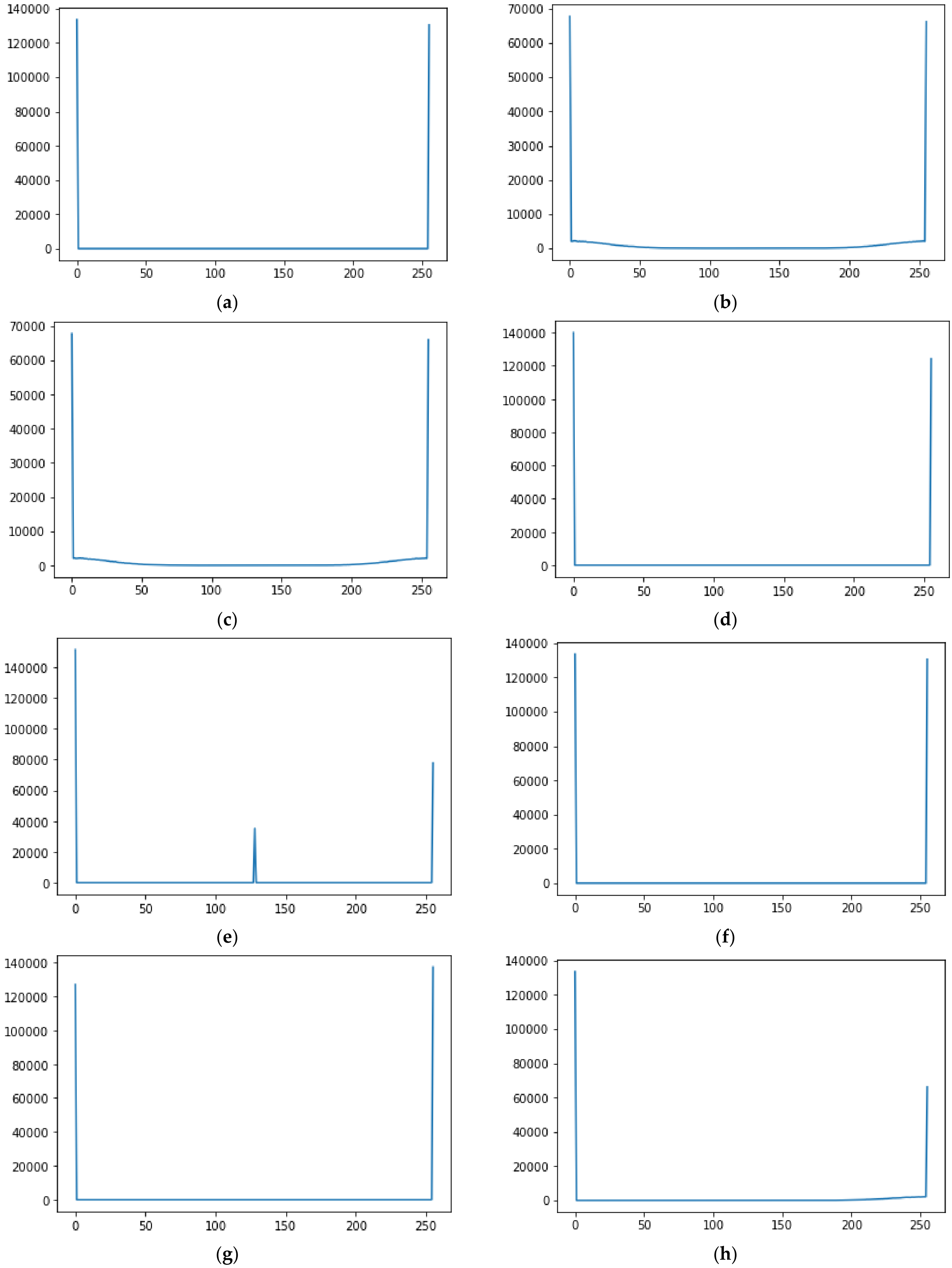


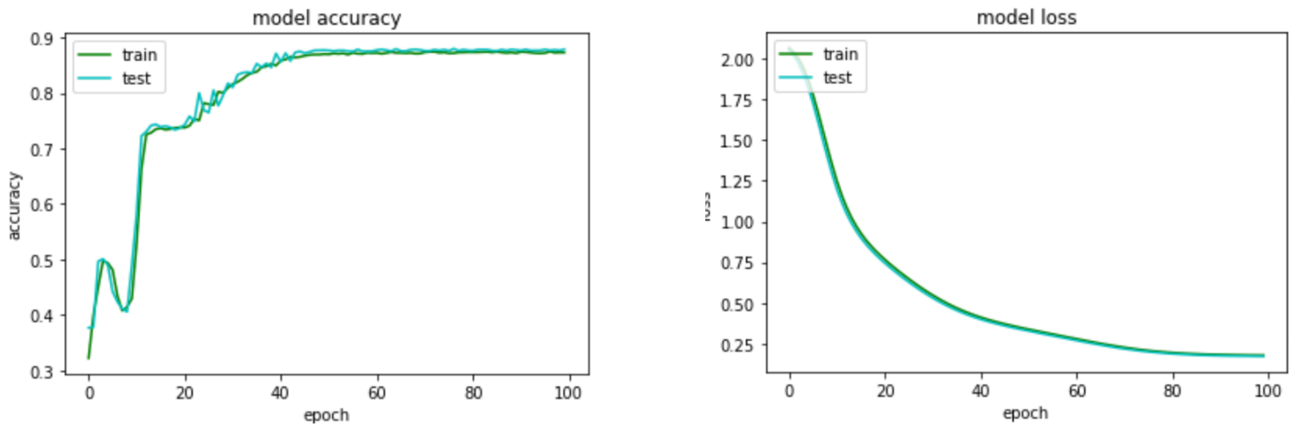
Figure 14. Histogram analysis of normal and noisy quick response (QR) code images where the x-axis represents the pixel number and the y-axis shows the value. (**a**) Normal QR codes; (**b**) Gaussian

noisy QR codes; (**c**) Localvar noisy QR codes; (**d**) Pepper noisy QR codes; (**e**) Poisson; (**f**) Salt and pepper noisy QR codes; (**g**) Salt noisy QR codes and (**h**) Speckle noisy QR codes.

4.2.1. Scenario 1 (8-Class Classification Task)

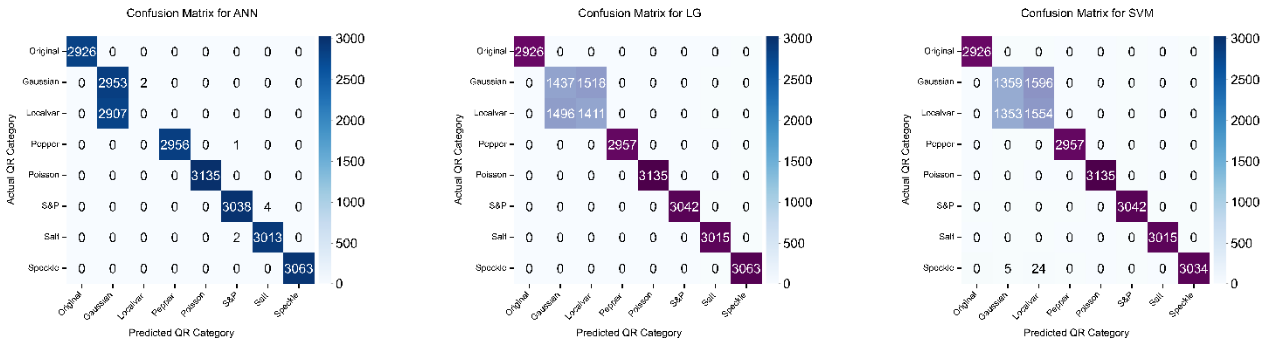
In this scenario, we utilized all eight types of QR code images, the original QR code and mapped seven noises (Gaussian, Localvar, Pepper, Poisson, Speckle, Salt, Salt and pepper) to the original images. The extracted features of images are fed to three classification models separately. All models are trained and tested over 70% and 30% of data, respectively.

Figure 15 depicts the performance curves of the proposed ANN model, whereas Figure 16 shows the confusion matrices for ANN, LG, and SVM. Experimental results show that all classification models (ANN, LG, and SVM) performed better when a hybrid scheme (combing histogram density feature extraction technique with machine learning classification algorithms) was exploited. All the models gained higher accuracy as compared to Scenario 1 in Scheme 1 of this paper. Table 9 depicts obtained performance measurements for each model in detail. However, from confusion matrices, it is noted that Localvar for the ANN model and Gaussian and Localvar for the other two models (LG and SVM) are not suitable.



(**a**) (**b**)

Figure 15. Performance curves of artificial neural network model when trained with features extraction through histogram density feature extraction technique to classify 8 types of quick response code images: (**a**) accuracy curves for train and test sets; (**b**) loss curves for train and test sets.



(**a**) (**b**) (**c**)

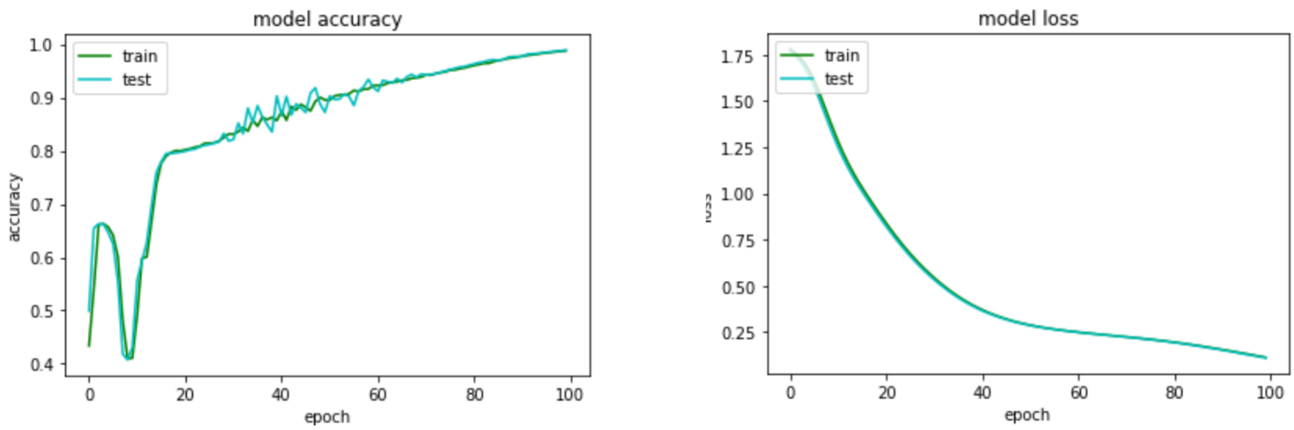
Figure 16. The confusion matrices for proposed classifiers when trained with features extraction through histogram density feature extraction technique to classify original quick response (QR) code and 7 different types of noisy QR code images. (**a**) artificial neural network (ANN); (**b**) logistic regression (LG); and (**c**) support vector machine (SVM).

Table 9. Performance comparison between artificial neural network (ANN), logistic regression (LG), and support vector machine (SVM) when trained with features extraction through histogram density feature extraction technique to classify 8 types of quick response code images.

Model	Accuracy	Precision	Recall	F1-score
ANN	**87.85%**	87.80%	87.30%	87.50%
LG	87.44%	87.37%	87.31%	87.34%
SVM	87.59%	87.15%	87.65%	87.40%

4.2.2. Scenario 2 (6-Class Classification Task)

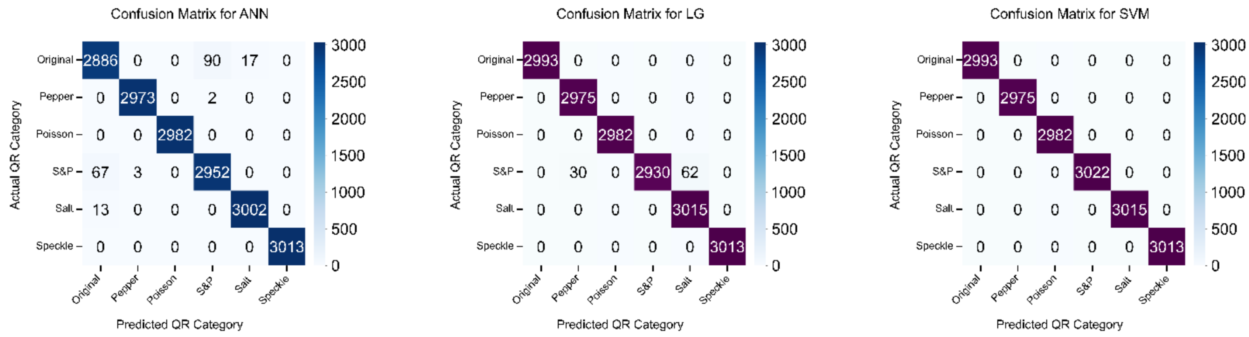
In this scenario, we removed two types of noisy images (Gaussian and localvar) and considered the rest of the six types of QR code images, the original QR code and noisy types of QR images (pepper, Poisson, speckle, salt, and salt and pepper). Similarly, the extracted features using the histogram density feature extraction technique are fed to three classification models separately. All models are trained and tested over 70% and 30% of data, respectively. Figure 17 depicts the performance curves of the proposed ANN model.



(**a**) (**b**)

Figure 17. Performance curves of artificial neural network model when trained with features extraction through histogram density feature extraction technique to classify six types of quick response code images: (**a**) accuracy curves for train and test sets; (**b**) loss curves for train and test sets.

Figure 18 shows the confusion matrices for ANN, LG, and SVM models, whereas Table 10 presents the overall performance of trained models. The exploited models performed better for six types of QR code image classification tasks. All the models gained accuracy as compared to Scenario 1. This time, SVM performed outstandingly by obtaining an overall accuracy of 100%, whereas LG reached 99.49% accuracy while ANN competes by accomplishing an overall accuracy of 98.93%. Moreover, the confusion matrices show that models performed well while segregating all six types of noisy QR codes.



(**a**) (**b**) (**c**)

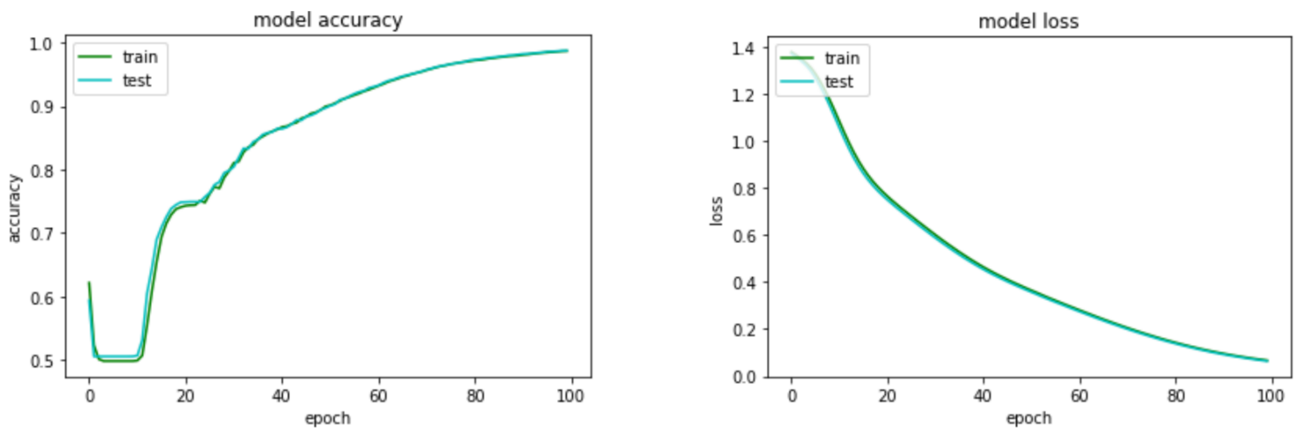
Figure 18. The confusion matrices for proposed classifiers when trained with features extraction through histogram density feature extraction technique to classify original quick response (QR) code and 5 different types of noisy QR code images. (**a**) artificial neural network (ANN); (**b**) logistic regression (LG); and (**c**) support vector machine (SVM).

Table 10. Performance comparison between artificial neural network (ANN), logistic regression (LG), and support vector machine (SVM) when trained with features extraction through histogram density feature extraction technique to classify six types of quick response code images.

Model	Accuracy	Precision	Recall	F1-score
ANN	98.93%	98.90%	98.90%	98.90%
LG	99.49%	99.50%	99.49%	99.49%
SVM	**100%**	100%	100%	100%

4.2.3. Scenario 3 (4-Class Classification Task)

In this scenario, we removed four types of noisy images (Gaussian, localvar, pepper, and speckle) and used the rest of the four types of QR code sets, the original QR code and three noisy types of QR code (Poisson, salt, and salt and pepper). The extracted features of images are fed to three classification models separately. All models are trained and tested over 70% and 30% of data, respectively. Figure 19 depicts the performance curves of the proposed ANN model.



(**a**) (**b**)

Figure 19. Performance curves of artificial neural network model when trained with features extraction through histogram density feature extraction technique to classify four types of quick response code images: (**a**) accuracy curves for train and test sets; (**b**) loss curves for train and test sets.

Figure 20 shows the confusion matrices for ANN, LG, and SVM models, while Table 11 presents the overall performance of trained models. The exploited classification models performed better for four types of QR code image classification tasks. All the models accomplished better accuracy as compared to Scenario 1 and Scenario 2. Moreover, the confusion matrices show that models are suitable for datasets composed of such four types of QR code images. However, SVM again outperformed the other two models by attaining full accuracy (100%).

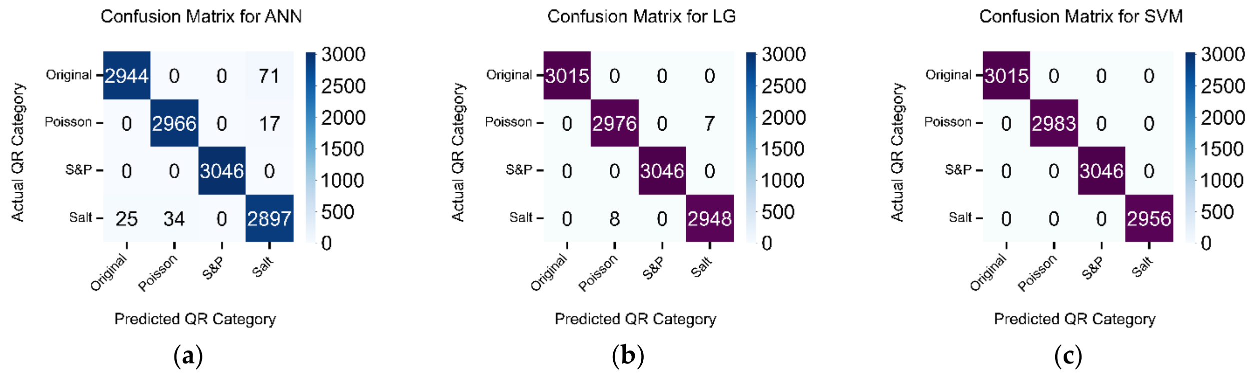


Figure 20. The confusion matrices for proposed classifiers when trained with features extraction through histogram density feature extraction technique to classify original quick response (QR) code and 3 different types of noisy QR code images. (**a**) artificial neural network (ANN); (**b**) logistic regression (LG); and (**c**) support vector machine (SVM).

Table 11. Performance comparison between artificial neural network (ANN), logistic regression (LG), and support vector machine (SVM) when trained with features extraction through histogram density feature extraction technique to classify four types of quick response code images.

Model	Accuracy	Precision	Recall	F1-score
ANN	98.77%	98.80%	98.80%	98.80%
LG	99.88%	99.87%	99.87%	99.87%
SVM	**100%**	100%	100%	100%

The experimental results prove that the hybrid approach (Scheme-2) is much more effective for the identification of noise types in QR code images. Furthermore, the introduction of training the models with different combinations of noisy (3-, 5-, and 7-type) QR code images shows that few noises share similar symmetrical characteristics that shape the image in such a way that implemented classifiers are unable to identify them correctly. For instance, the classifiers in both the schemes (I and II) could not classify the QR code images properly that have Gaussian noise and Localvar noises. The introduction of the histogram density feature extraction technique significantly enhanced the performance of each classifier in all three scenarios, as shown in Table 12. It is noted that the histogram density analysis technique shaped each QR code imaging data into 256 useful features that dramatically boost the learning of the model. Among all, SVM with histogram density feature extraction technique performed well as it attained the highest performance while classifying five types and three types of noisy images along with original QR code images.

All the experiments are carried out on Intel® Core™ i9-10900KF CPU 3.70 GHz, 64GB RAM with NVIDIA GeForce RTX 3070 GPU. The system took 223 min to train the ANN model along the histogram density feature extraction technique; however, for LG and SVM, it took 201 and 210 min, respectively.

Table 12. Comparative performance analysis of proposed convolutional neural network (CNN), artificial neural network (ANN), logistic regression (LG), and support vector machine (SVM) when trained with/without features extraction through histogram density feature extraction (HDFE) technique to classify 8-, 6-, and 4-types of quick response code images.

Classification Problem	Model	Accuracy	
		Without HDFE	With HDFE
8-class	CNN/ANN	85.60%	87.85%
	LG	58.85%	87.44%
	SVM	69.74%	87.59%
6-class	CNN/ANN	97.70%	98.93%
	LG	77.48%	99.49%
	SVM	90.53%	**100%**
4-class	CNN/ANN	98.43%	98.77%
	LG	98.09%	99.88%
	SVM	98.91%	**100%**

5. Conclusions

The printing, scanning, and transmission of QR codes may introduce noise and corrupt the information. However, a message or information can be recovered if the noise type is identified. Therefore, this paper proposed an amalgam approach based on computer vision techniques and a machine learning classification algorithm to identify the type of noise present in QR code images. To investigate the proposed approach, we generated a dataset of 80,000 images that contain images of the original QR code and seven various types of noisy QR code images. The noises include salt and pepper, pepper, speckle, Poisson, salt, localvar, and Gaussian. First, we analyzed the performance of several machine learning and deep learning classifiers by directly feeding the generated dataset images and observed that the models could not successfully segregate the noisy images. Later, a histogram density analysis technique is incorporated to extract the useful features by shaping the data into a more representable form and then fed to the artificial neural network (ANN), support vector machine (SVM), and logistic regression (LG), separately. It is observed that the classification performance improved significantly with the introduction of histogram density feature extraction techniques. Moreover, the impact of introducing various noises in QR code images on classification performance is also provided by training the models in three different scenarios (where original images and combinations of three, five, and seven types of noisy images are used). SVM with histogram density analysis performed well by attaining 100% accuracy for 6-class (original, pepper, Poisson, speckle, salt, and salt and pepper) and 4-class (original, Poisson, salt, and salt and pepper) classification tasks. However, in the future, the approach can be enhanced to cater the classification of all seven noises along with the original QR code image, as currently, it hardly attains 88% approximately.

Author Contributions: Conceptualization, J.R. and M.Y.; methodology, J.R., A.B.W. and M.Y.; software, A.M.A.-M., T.U. and S.W.; validation, J.R., A.B.W. and S.W.; formal analysis, J.R., A.B.W. and M.Y.; investigation, J.R., A.B.W., A.M.A.-M. and T.U.; resources, J.R.; data curation, J.R., A.B.W. and M.Y.; writing—original draft preparation, J.R., A.B.W. and S.W.; writing—review and editing, J.R.; visualization, A.M.A.-M., T.U. and S.W.; supervision, J.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: As our institute reserves the rights to the generated data, it is not publicly available, but can be provided to an individual upon request. Requests can be made by emailing the corresponding author.

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

References

1. Zhang, X.; Duan, J.; Zhou, J. A Robust Secret Sharing QR Code via Texture Pattern Design. In Proceedings of the 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC); IEEE: Honolulu, HI, USA, Nov. 2018; pp. 903–907.
2. International Organization for Standardization. *International Organization for Standardization: Information Technology—Automatic Identification and Data Capture Techniques—Bar Code Symbology—QR Code; ISO*; June 2000.
3. Chen, J.; Huang, B.; Mao, J.; Li, B. A Novel Correction Algorithm for Distorted QR-code Image. In Proceedings of the 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE); IEEE: Xiamen, China, Oct. 2019; pp. 380–384.
4. Lee, J.-K.; Wang, Y.-M.; Lu, C.-S.; Wang, H.-C.; Chou, T.-R. The Enhancement of Graphic QR Code Recognition using Convolutional Neural Networks. In Proceedings of the 2019 8th International Conference on Innovation, Communication and Engineering (ICICE); IEEE: Zhengzhou, China, Oct. 2019; pp. 94–97.
5. Hosseini, H.; Hessar, F.; Marvasti, F. Real-Time Impulse Noise Suppression from Images Using an Efficient Weighted-Average Filtering. *IEEE Signal Process. Lett.* **2015**, *22*, 1050–1054. https://doi.org/10.1109/LSP.2014.2381649.
6. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data* **2022,** *8, 53*. https://doi.org/10.1186/s40537-021-00444-8. Publishing: 2021;
7. Mishra, D.; Singh, S.K.; Singh, R.K. Deep Architectures for Image Compression: A Critical Review. *Signal Process.* **2022**, *191*, 108346. https://doi.org/10.1016/j.sigpro.2021.108346.
8. Al-Saffar, A.A.M.; Tao, H.; Talab, M.A. Review of deep convolution neural network in image classification. In Proceedings of the 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET); IEEE: Jakarta, Indonesia, Oct. 2017; pp. 26–31.
9. Yu, S.; Jia, S.; Xu, C. Convolutional neural networks for hyperspectral image classification. *Neurocomputing* **2017**, *219*, 88–98. https://doi.org/10.1016/j.neucom.2016.09.010.
10. Yim, J.; Sohn, K.-A. Enhancing the Performance of Convolutional Neural Networks on Quality Degraded Datasets. In Proceedings of the 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA); IEEE: Sydney, NSW, Australia, Nov. 2017; pp. 1–8.
11. Zou, X.; Hu, Y.; Tian, Z.; Shen, K. Logistic Regression Model Optimization and Case Analysis. In Proceedings of the 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT); IEEE: Dalian, China, Oct. 2019; pp. 135–139.
12. Thai, L.H.; Hai, T.S.; Thuy, N.T. Image Classification using Support Vector Machine and Artificial Neural Network. *Int. J. Inf. Technol. Comput. Sci.* **2012**, *4*, 32–38. https://doi.org/10.5815/ijitcs.2012.05.05.
13. Nikolaev, A.V.; de Jong, L.; Weijers, G.; Groenhuis, V.; Mann, R.M.; Siepel, F.J.; Maris, B.M.; Stramigioli, S.; Hansen, H.H.G.; de Korte, C.L. Quantitative Evaluation of an Automated Cone-Based Breast Ultrasound Scanner for MRI–3D US Image Fusion. *IEEE Trans. Med. Imaging* **2021**, *40*, 1229–1239. https://doi.org/10.1109/TMI.2021.3050525.
14. Rasheed, J. Analyzing the Effect of Filtering and Feature-Extraction Techniques in a Machine Learning Model for Identification of Infectious Disease Using Radiography Imaging. *Symmetry* **2022**, *14*, 1398. https://doi.org/10.3390/sym14071398.
15. Pandey, R.C.; Singh, S.K.; Shukla, K.K. Passive forensics in image and video using noise features: A review. *Digit. Investig.* **2016**, *19*, 1–28. https://doi.org/10.1016/j.diin.2016.08.002.
16. Sil, D.; Dutta, A.; Chandra, A. Convolutional Neural Networks for Noise Classification and Denoising of Images. In Proceedings of the TENCON 2019–2019 IEEE Region 10 Conference (TENCON); IEEE: Kochi, India, Oct. 2019; pp. 447–451.
17. Momeny, M.; Latif, A.M.; Agha Sarram, M.; Sheikhpour, R.; Zhang, Y.D. A noise robust convolutional neural network for image classification. *Results Eng.* **2021**, *10*, 100225. https://doi.org/10.1016/j.rineng.2021.100225.
18. Roy, S.S.; Ahmed, M.U.; Akhand, M.A.H. Noisy Image Classification Using Hybrid Deep Learning Methods. *J. Inf. Commun. Technol.* **2018**, *17*, 233–269. https://doi.org/10.32890/jict2018.17.2.8253.
19. Khaw, H.Y.; Soon, F.C.; Chuah, J.H.; Chow, C. Image noise types recognition using convolutional neural network with principal components analysis. *IET Image Process.* **2017**, *11*, 1238–1245. https://doi.org/10.1049/iet-ipr.2017.0374.
20. Geng, L.; Zicheng, Z.; Qian, L.; Chun, L.; Jie, B. Image Noise Level Classification Technique Based on Image Quality Assessment. In Proceedings of the 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS); IEEE: Shenyang, China, July 2020; pp. 651–656.
21. Tripathi, M. Facial image noise classification and denoising using neural network. *Sustain. Eng. Innov.* **2021**, *3*, 102–111. https://doi.org/10.37868/sei.v3i2.id142.

22. Prakash, K.B.; Ruwali, A.; Kanagachidambaresan, G.R. Introduction to Tensorflow Package. In: Prakash, K.B., Kanagachidambaresan, G.R. (eds) Programming with TensorFlow. EAI/Springer Innovations in Communication and Computing. Springer, Cham. 2021; pp. 1–4.
23. Barbu, T. Variational Image Denoising Approach with Diffusion Porous Media Flow. *Abstr. Appl. Anal.* **2013**, *2013*, 1–8. https://doi.org/10.1155/2013/856876.
24. *The Essential Guide to Image Processing*, 1st ed.; Bovik, A., Eds.; Elsevier: Amsterdam, The Netherlands, 2009; ISBN 9780123744579.
25. Nath, A. Image Denoising Algorithms: A Comparative Study of Different Filtration Approaches Used in Image Restoration. In Proceedings of the 2013 International Conference on Communication Systems and Network Technologies; IEEE: Gwalior, India, April 2013; pp. 157–163.
26. Talbot, H.; Phelippeau, H.; Akil, M.; Bara, S. Efficient Poisson denoising for photography. In Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP); IEEE: Cairo, Egypt, Nov. 2009; pp. 3881–3884.
27. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory—COLT '92; ACM Press: July 1992; pp. 144–152.
28. Akbari, A.; Awais, M.; Bashar, M.; Kittler, J. How Does Loss Function Affect Generalization Performance of Deep Learning? Application to Human Age Estimation. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning; Meila, M., Zhang, T., Eds.; PMLR: July 2021; Volume 139, pp. 141–151. https://proceedings.mlr.press/v139/akbari21a.html.