Noise Presence Detection in QR Code Images

Conference Paper · October 2022

DOI: 10.1109/ACIT54803.2022.9912751

6 authors, including:

Sadaf Waziry
Istanbul Aydın University

Jawad Rasheed
Istanbul Sabahattin Zaim University

Amani Yahyaoui
Sakarya University

Mirsat Yesiltepe
Yıldız Technical University

Noise Presence Detection in QR Code Images

Ahmad Bilal Wardak
Department of Software Engineering
Istanbul Aydin University
Istanbul, Turkey
0000-0002-7928-5234

Jawad Rasheed
Department of Computer Engineering
Istanbul Aydin University
Istanbul, Turkey
0000-0003-3761-1641

Amani Yahyaoui
Department of Software Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-003-0603-6592

Sadaf Waziry
Department of Software Engineering
Istanbul Aydin University
Istanbul, Turkey
0000-0001-5082-4794

Erdal Alimovski
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul, Turkey
0000-0003-0909-2047

Mirsat Yesiltepe
Dept.. of Mathematical Engineering
Yildiz Technical University
Istanbul, Turkey
0000-0003-4433-5606

2022 12th International Conference on Advanced Computer Information Technologies (ACIT) | 978-1-6654-1050-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ACIT54803.2022.9912751

***Abstract*—A quick response (QR) code is symbols used to encode information such as key identifiers (website addresses, product, etc.) that can be printed and scanned electronically using image-based technology. However, it may include noise at the time of printing or scanning due to some environmental or mechanical factors. Therefore, the study analyzes various machine learning models to detect noise presence in QR code. For this, we first generated own dataset by creating 14,000 images of QR code, and then enhanced the dataset by adding several noises to the original QR code images. Later, it exploits several machine learning, deep learning and pre-trained models to segregate noisy images from original images. Experimental results show that ResNet101 and Xception models outperformed others by attaining 100% accuracy, recall, f1-score, and precision, each. Besides these, support vector machine (SVM) also performed better by accomplishing 99.6% accuracy on test set when trained over 70% of dataset.**

Keywords—noise detection, machine learning, deep learning

I. INTRODUCTION

A quick response (QR) code is a machine-readable optical label that carries information about the associated item or product [1]. Barcodes encode information in just one vector. On the contrary, information in a two-dimensional code, such as the QR code, is coded in both horizontal and vertical directions. It is easy to read and can hold a big quantity of data. In 1994, Denso Wave Incorporated in Japan devised the QR code [2]. It was thereafter frequently employed as a distinguishing mark for a variety of commercial objects, marketing, and public statements.

Various noises, including Poisson, Gaussian, Speckle, Salt & Pepper and others, may lower the quality of a QR code image throughout digital image acquisition and transmission [3]. These noises are caused by poor environmental factors. Noisy QR code images are harmful to model training, decreasing network classification performance. As a result, better approaches for differentiating between original and noisy QR Codes are needed.

Mmachine learning classical algorithms such as Support Victor Machine (SVM) [4], Decision Tree (DT), Random Forest (RF), K-Nearest Neighbor (KNN) [5] and etc. are very trendy in the technology industry [6] and they are being used in solving different kind of classification problems. Deep learning has also been shown encouraging results in a variety of applications for artificial intelligence, including image identification, natural language processing, language modeling, and other similar applications [7] and it is a subfield of machine learning [8] and one of the most significant achievements and research hotspots in recent times. CNN is a kind of deep neural network that handles feature detection by varying between dozens or even hundreds of hidden layers. This allows us to build deep layers and manage a large variety of image input attributes [9]. It is a powerful method for classifying images because of the increasing complexity of the collected attributes with each additional layer.

A number of studies have made an effort to classify different types of visual noise in images. For instance, Liu et al. [10] use two kinds of deep learning networks: VGG16-based CNN networks and feed-forward CNN networks to classify the noise type of the images and denoising them based on the classification result. With these efforts, their system can automatically denoise single or mixed forms of noise. Their experimental results reveal that the classification network outperforms the current approaches in terms of accuracy, while the denoising network outperforms them in terms of PSNR and SSIM.

To eliminate label noise and its negative effects on deep neural network training, it is essential to develop counter algorithms; thus, the paper [11] intends to present several algorithms and classify them into two different groups: noise model-based and noise model-free methods. The first set of algorithms is to assess the noise structure and utilize this knowledge to prevent the negative consequences of noisy labeling. Methods in the second category, on the other hand, attempt to develop intrinsically noise-resistant algorithms by using methodologies such as resilient losses, regularizers, or other learning methodologies.

It was illustrated in paper [12] how to identify image noise of varying types and relative intensity using a backpropagation algorithm, CNN method and stochastic gradient descent optimization technique. Paper [13] utilized two different CNNs, Inception-v3 and VGG-16, to clearly detect noise representations. Milan Tripathi [14] developed, deployed, and analyzed a CNN-based classification model to identify noisy images with high validation and training validity. Additionally, he developed a UNET-based framework to denoise images with optimum peak-signal-to-noise ratio and a method to accurately measure values.

When printing the QR Code, we are faced with a noise phenomenon that prevents the code from being recognized and ultimately results in failure. As a consequence of this, we are investigating the possibility of developing an intelligent image categorization technique for both noisy and original varieties of QR codes. Although the ability to eliminate noise is critical, it is equally necessary to evaluate whether the QR code image includes noise or is an authentic one. We offered

CNN, SVM, and deep learning pre-trained models (Xception and RestNet101) to effectively detect the original and noisy QR Code images to tackle this challenge. Also, we have used Naïve Bayes, Decision Tree, Random Forest algorithms as a comparison to the mentioned methods result. The paper is further broken into parts as follows: The recommended method is broken out in more detail in Section 2. The results of the experiments are discussed in Section 3, along with a comparison and a discussion of the results. In Section 4, the conclusion is discussed.

II. PROPOSED METHOD

The aim of this work is to develop a classification model that, given images of QR codes as input, can determine if those images include genuine QR codes or QR codes that are noisy. This research generated its own collection of QR code images and applied numerous noises (Salt, and Paper Gaussian, Speckle, Poisson, Speckle) to the original images.

The main processes of this study are to conduct an analysis of QR code images that have been generated, resize the images, map noise to the original QR code images, encode the labels, train the proposed CNN, SVM, and pre-trained (Xception, ResNet101) models, identify the type of QR code image, output the classified category regardless of whether the image is the original QR code image or a noisy one, and compare the results of the models with the results gained from NB, DT and RF algorithms.

A. Data Analyzation

The dataset that we are employing for this study consists of 40000 images, including both original and noisy versions of the original QR Code. The dataset is rather diverse; since our models need the input of a given size, we pre-processed the images to shrink them into the fixed size of (160x160) ratio for the CNN, Xception, and ResNet101 models and (50x50) ratio for the SVM, NB, DT, and RF models. This allowed our models to accurately analyze the data. Fig. 1 displays the original image along with many different forms of noise that were added to the image. Our data set is partitioned into two unique classes: the first is the "original" type, and the second is the "noisy" type. To get the noisy type, we combined the "original" type with four various forms of noise, including Speckle, Gaussian, Poisson and Salt & Pepper.

B. CNN Model Architecture

In order to categorize the original as well as the noisy QR Code images, we developed a CNN model. It is constructed using five layers of convolutional (16, 32, 64, 128, and 128 kernels respectively), each followed by max-pooling layer. At least, it has a dropout layer, flattening layer, and one layer of fully connected data. Network is trained using Adam optimizer, having a batch size of 120 over 100 epochs.

C. SVM Model

SVM is indeed an effective pattern recognition and image classification algorithm [4]. SVM builds the optimum separating hyperplanes based on a kernel function. Every piece of data in the SVM technique is plotted as a point in n-dimensional space, where n represents the size of features, and the value of each feature is the value of a specific coordinate. We first scaled and reshaped our data in the suggested SVM model, then used the Standard Scaler to reshape the distribution of values such that the mean of the observed values is 0 and the standard deviation is 1. Such that, the mean is removed and each feature is scaled to unit variance..

D. Xcpetion Model

Xception model [15] feature extraction foundation is made up of 71 convolutional layers. Xception's layers, with the exception of the first and final modules, are structured into modules that are all encircled by linear residual connections. The general design of Xception is comprised of three flows: entrance, middle, and exit flow. Data is transferred via the input flow first, then the middle flow, which repeats itself eight times, and finally the exit flow.

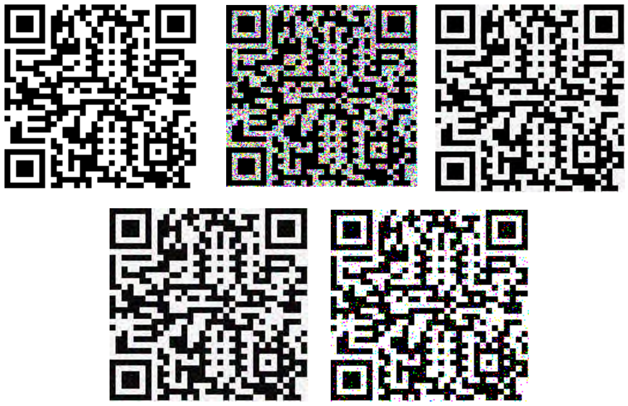

Fig. 1. A representation of QR Code images, including the original image as well as four different forms of mapped noise in the form of noisy images.

E. ResNet101 Model

ResNet-101 is a 101-layer CNN. Residual Networks, often known as ResNets, are a form of classic neural network that serves as the basis for many computer vision applications [16]. ResNet pioneered the concept of skip connections, which alleviates the problem of vanishing gradient by allowing the gradient to stream through a replacement shortcut manner and permitting the model to learn an identity mapping that guarantees the higher layer performs well. ResNet's key breakthrough was that it allowed us to train extraordinarily deep neural networks with 150+ layers.

In order to perform QR Code classification using these two pre-trained networks, we made some adjustments to the models so that they could be trained using our dataset. Because our data will be divided into two distinct groups—namely, Original and Noisy—we need to add one more layer called Dense that has two outputs.

III. RESULT AND DISCUSSION

To determine how accurate our models are, we utilized a dataset that included 40,000 images of QR codes with four different noise types. The dataset was split into the train set and test set using the proportions 70/30 respectively. Table I contains more information on the partitioning of the dataset. After the training was completed successfully, the accuracy was determined by applying each and every image from the test dataset in every iteration. The accuracy and loss of the CNN, Xception, and ResNet101 models during training are shown in Fig. 2 – 4, respectively. Accuracy, f1-score, recall and precision were the four metrics that were used in order to assess the utility and performance of our models. Table II – V provide the results of the metric calculations that were carried out on the test data after the training of the CNN, Xception, ResNet101 and SVM models correspondingly. Fig. 5 presents a representation of the confusion matrix that was derived from the testing data for all QR Code pictures using the SVM model.

TABLE I. INFORMATION OF QR CODE IMAGE DATASET

Label/Class	Training Set	Testing Set	Total
Normal/Original QR	14000	6000	20000
Poisson	3500	1500	5000
Speckle	3500	1500	5000
Salt & Pepper	3500	1500	5000
Gaussian	3500	1500	5000
Total	**28000**	**12000**	**40000**

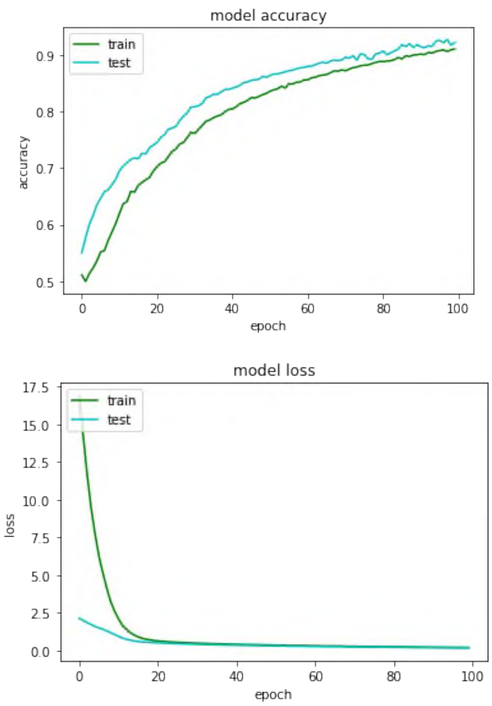

Fig. 2. Proposed CNN model training and validation data accuracy improvement and loss reduction

TABLE II. PERFORMANCE METRICS OF SUGGESSTED CNN MODEL

Model	Accuracy	F1-Score	Recall	Precision
CNN	92.15%	92.10%	92.10%	92.10%

TABLE III. PERFORMANCE METRICS OF PROPOSED XCEPTION MODEL

Model	Accuracy	F1-Score	Recall	Precision
Xception	100%	100%	100%	100%

TABLE IV. PERFORMANCE METRICS OF PROPOSED RESNET101 MODEL

Model	Accuracy	F1-Score	Recall	Precision
ResNet101	100%	100%	100%	100%

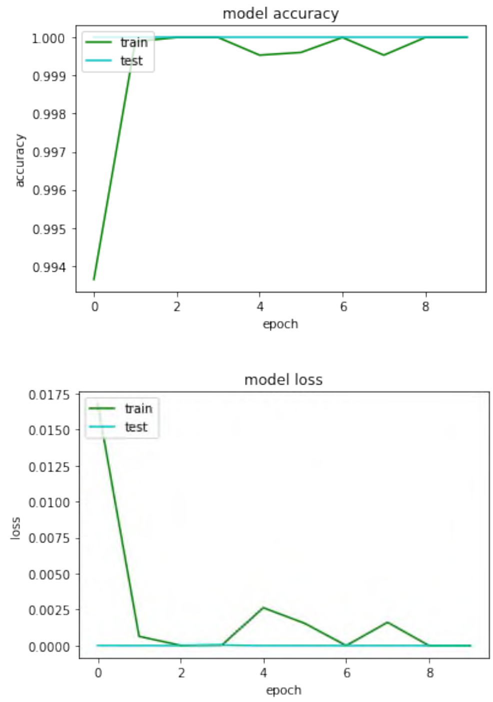

Fig. 3. Proposed Xception model training and validation data accuracy improvement and loss reduction

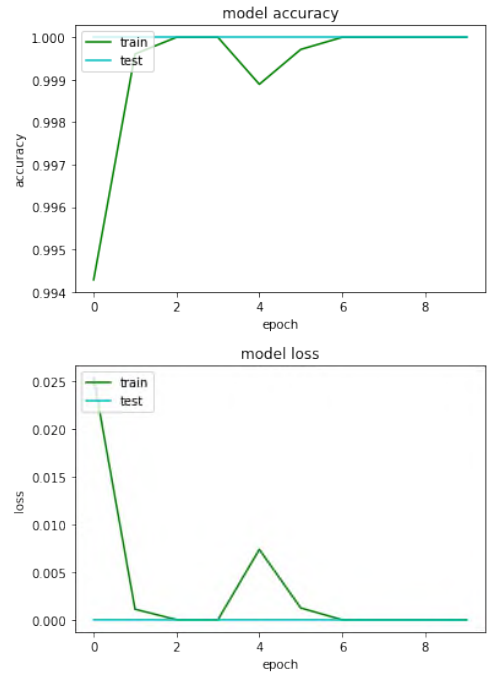

Fig. 4. Proposed ResNet101 model training and validation data accuracy improvement and loss reduction

TABLE V. PERFORMANCE METRICS OF PROPOSED SVM MODEL

Model	Accuracy	F1-Score	Recall	Precision
SVM	99.60%	99.61%	99.61%	99.61%

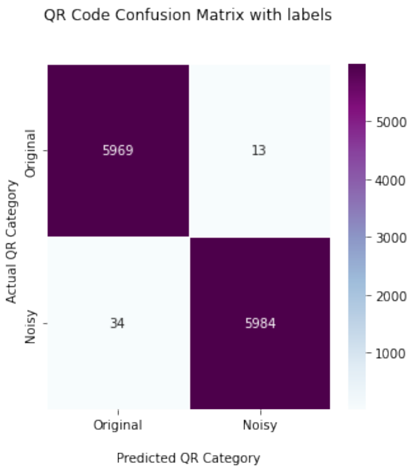

Fig. 5. Confusion matrics of proposed SVM model with the test data

From the above results we can see that among all the models the deep learning pre-trained models, Xcpetion and ResNet101 outperform the CNN and SVM models with gaining the accuracy of 100%. To have a more accurate and effective QR Code noise classification we also feed the dataset to other machine learning classical algorithms such as Random Forest, Decision Tree, and Naïve Bayes and compared the results with the SVM and deep learning models **Table VI**. And the overall result shows that among classical machine algorithms the SVM model works more effective for classifying the QR code images noise and from deep learning models the pre-trained models are the efficient ones.

TABLE VI. PERFORMANCE METRICS OF NB, RF, CNN, DT, SVM, XCEPTION AND RESTNET101MODELS

Model	Accuracy	F1-Score	Recall	Precision
Naïve Bayes	74.60%	72.85%	74.60%	83.15%
Random Forest	91.80%	91.75%	91.78%	92.97%
CNN	92.15%	92.10%	92.10%	92.10%
Decision Tree	93.43%	93.41%	93.47%	94.17%
SVM	**99.60%**	**99.61%**	**99.61%**	**99.61%**
Xception	**100%**	**100%**	**100%**	**100%**
ResNet101	**100%**	**100%**	**100%**	**100%**

IV. CONCLUSION

This study exploited traditional machine learning classifiers (Naïve Bayes, random forest, decision tree, and support vector machine), deep learning algorithms (convolutional neural network) and pre-trained models (Xception and ResNet101) to identify either the image containing QR code has noise in it or not. In order to investigate the performance of these models, the study generated a new dataset containing 40,000 images comprised of original QR code and noisy QR code images. Dataset generation includes integration of several types of noises (Poisson, speckle, salt & pepper, and Gaussian). Extensive experiments were carried out to train the models by splitting the dataset into 70/30 ratio for train/test set. Among all SVM, Xception, and ResNet101 performed better by attaining an overall accuracy of 99.6%, 100% and 100%, respectively. The future work includes the classification of noise types in a given image.

REFERENCES

[1] J. H. Chang, "An introduction to using QR codes in scholarly journals," *Sci. Ed.*, vol. 1, no. 2, pp. 113–117, 2014, doi: 10.6087/kcse.2014.1.113.

[2] J. Chen, B. Huang, J. Mao, and B. Li, "A novel correction algorithm for distorted QR-code image," *2019 IEEE 3rd Int. Conf. Electron. Inf. Technol. Comput. Eng. EITCE 2019*, no. 2017, pp. 380–384, 2019, doi: 10.1109/EITCE47263.2019.9095073.

[3] H. Hosseini, F. Hessar, and F. Marvasti, "Real-time impulse noise suppression from images using an efficient weighted-average filtering," *IEEE Signal Process. Lett.*, vol. 22, no. 8, pp. 1050–1054, 2015, doi: 10.1109/LSP.2014.2381649.

[4] L. H. Thai, T. S. Hai, and N. T. Thuy, "Image Classification using Support Vector Machine and Artificial Neural Network," *Int. J. Inf. Technol. Comput. Sci.*, vol. 4, no. 5, pp. 32–38, 2012, doi: 10.5815/ijitcs.2012.05.05.

[5] R. Udaiyakumar, N. Vijayalakshmi, M. Prashanthram, and S. Jayaprakash, "A Comparative Study on Machine Learning and Artificial Neural Networking Algorithms," *2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020*, pp. 516–517, 2020, doi: 10.1109/ICACCS48705.2020.9074203.

[6] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021.

[7] P. P. By Mohit Sewak , Md. Rezaul Karim, *Practical Convolutional Neural Networks*. Packt, 2018.

[8] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, "Review of deep convolution neural network in image classification," *Proceeding - 2017 Int. Conf. Radar, Antenna, Microwave, Electron. Telecommun. ICRAMET 2017*, vol. 2018-Janua, no. October, pp. 26–31, 2017, doi: 10.1109/ICRAMET.2017.8253139.

[9] J. Yim and K. A. Sohn, "Enhancing the Performance of Convolutional Neural Networks on Quality Degraded Datasets," *DICTA 2017 - 2017 Int. Conf. Digit. Image Comput. Tech. Appl.*, vol. 2017-Decem, pp. 1–8, 2017, doi: 10.1109/DICTA.2017.8227427.

[10] F. Liu, Q. Song, and G. Jin, "The classification and denoising of image noise based on deep neural networks," *Appl. Intell.*, vol. 50, no. 7, pp. 2194–2207, 2020, doi: 10.1007/s10489-019-01623-0.

[11] G. Algan and I. Ulusoy, "Image classification with deep learning in the presence of noisy labels: A survey," *Knowledge-Based Syst.*, vol. 215, no. 89, 2021, doi: 10.1016/j.knosys.2021.106771.

[12] H. Y. Khaw, F. C. Soon, J. H. Chuah, and C. O. Chow, "Image noise types recognition using convolutional neural network with principal components analysis," *IET Image Process.*, vol. 11, no. 12, pp. 1238–1245, 2017, doi: 10.1049/iet-ipr.2017.0374.

[13] D. Sil, A. Dutta, and A. Chandra, "Convolutional Neural Networks for Noise Classification and Denoising of Images," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2019-Octob, pp. 447–451, 2019, doi: 10.1109/TENCON.2019.8929277.

[14] M. Tripathi, "Facial image noise classification and denoising using neural network," *Sustain. Eng. Innov.*, vol. 3, no. 2, pp. 102–111, 2021, doi: 10.37868/sei.v3i2.id142.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.