# sQR: High Performance Secure QR System to Filter against Malicious QR Codes using Deep Learning Techniques

Sanchit Goel, Gourab Chakraborty, Nihar Sanda, Manjunath K Vanahalli

[a]*Indian Institute of Information Technology Dharwad*
[b]*{19bcs095, 19bcs118, 19bcs125, manjunathkv}@iiitdwd.ac.in,*

**Abstract**

With the tremendous growth in Quick Response Code (QR Code) adoption in emerging markets for financial transactions, contact tracing, contactless registration, user authentication and verification, the amount of QR code related frauds have also risen significantly. Without effectively tackling the problem of malicious QR codes being used by hackers, we can't truly unlock the potential of QR codes in the digital transformation of emerging markets. QR is a 2D barcode that can encode multiple data types which can be accessed by digital devices like smartphones that have a camera embedded in it to scan the QR. Most of these malicious QR code detection systems are dependent on advanced cryptographic techniques that are implemented after decoding the QR to its corresponding link. This process takes up a significant amount of time. The solution proposed in this paper aims to solve the problem of malicious QR code detection in a very fast and efficient manner taking advantage of state-of-the-art deep learning models and pattern recognition techniques. A dataset of 200000 malicious and benign QR code images was used to train our deep learning models. An in-depth analysis of how information is encoded in a QR code is performed before applying deep learning models. On implementing Visual Geometry Group's VGG16, Efficient Net, which are Convolutional Neural Network (CNNs), and Vision Transformer, the transformer was found to be the most effective at 99.965% accuracy in identifying malicious QR codes. Based on the outcomes, a secure QR (sQR) has been proposed in this paper which securely detects malicious QR codes and alerts the user.

*Keywords:* Deep Learning, Security, Pattern Recognition, QR code

*PACS:* 0000, 1111
*2000 MSC:* 0000, 1111

## 1. Introduction

QR code is a matrix barcode consisting of an array of white and black squares which is a machine readable label that holds the details relevant to the product it is attached to, generally via an URL. Scanning the QR code opens up the link to a website or redirects to an application. The retrieval of the data encoded in a QR code occurs within a few seconds due to the ultrahigh speed used to verify the validity of the code received from the sensor. QR codes use the following encoding methods:

1. binary/byte
2. alphanumeric
3. numeric

Other than that, there are several extensions available on top of these basic encoding methods.

Researchers directed their attention to QR codes since it is more economical and accessible due to the smartphones' cameras. It provided a better alternative identification source as compared to fingerprints and barcodes. QR codes were confirmed as an international standard in 2000. The current standard version was published in 2015.

The encoding and function pattern sections make up the majority of QR code images. Due to the linear (one-dimensional, 1D) restricted technological capabilities of barcode, QR codes have been widely deployed. However, there has been a rising need for information storage that goes beyond what a 1D barcode can offer. In several industries, QR codes are now commonplace. They are used in the fields of education, transportation, product tracking, ticketing, smarttags, book return procedures in libraries, payment transfer systems, and tourism promotion and can be mounted to any screen, poster, or product surface. In several industries today, including education, fish farming, land management, and healthcare services, QR codes are employed to ensure a sustainable marketing strategy [1]. For instance, in this setting, QR codes can enhance healthcare services by managing patient identification effectively. The wristbands' QR codes can be linked to personal information. To access patient information, medicines, and medical reports, healthcare

providers can utilize a smartphone app that scans QR codes. High-speed component scanning in factories is made possible by QR codes. In some circumstances, secure barcodes can be used as a cost-effective substitute for Radio Frequency IDentifier (RFID) tags in Internet of Things (IoT) programmes to provide security, privacy, and management levels. Barcodes can operate as a link between IoT gadgets and the cloud, which can manage huge data operations and allow security considerations in IoT development. QR codes are associated with a number of security risks. Without a certain reader device or applications, barcodes cannot be read. However, there is no recognised methodology for creating QR codes, therefore they could be vulnerable to a number of dubious assaults. There are several security safeguards in place in case barcodes are used in QR code assaults. Some of these solutions are only applicable to malicious link detection technologies that are familiar with cryptography techniques.

The main objective of this paper is to identify if the QR code is malicious or not without decoding the underlying URL that is embedded in the QR code. A dataset of 200,000 malicious and benign QR Codes was created using a python script that generated QR Codes out of a verified dataset containing a list of live URLs labeled malignant or benign. Furthermore, three deep learning models were built and compared to select the most suitable model for detecting malicious QR codes. Two models are based on CNN architecture (VGG16 and EfficientNet) whereas one model is based on transformer architecture (Vision Transformer).

More precisely, the specific contributions of this paper are summarized as follows

1. Built an efficient deep learning model on a robust and exhaustive dataset for classifying malicious QR codes with high accuracy.
2. Comparative Study on various Deep Learning models on the basis of scoring metrics.
3. Developed a secure system integrating the model demonstrating the same can be used by other QR based applications and protocols.
4. Developed Open REST API's for integration into third-party applications as demonstrated in the Proof of Concept (PoC).

## 2. Literature Review

The literature review discussed in this section ascertains the state-of-the-art research on the available counter-measures and solutions to preserve 2D

barcodes.

The study initially provides a glossary of terms and algorithms related to information security and cryptography in this section. Then, we'll talk about security options for barcodes. The three concepts known as the CIA triad (Confidentiality, Integrity, and Availability) make up the majority of security jargon. Data security refers to preventing unauthorized parties from accessing it. Data is typically encrypted to make it so that only people with the proper permissions and access to the key may decrypt it and view the contents. Assuring the accuracy of data delivery and that it was not altered by unauthorized parties is part of maintaining data integrity. Additionally, availability means that the information system must be accessible at all times. Information security also comprises non repudiation, which makes sure that an entity cannot dispute the transmission of a message or the signing of a document, and authentication, which tries to confirm the identity of individuals or entities.

Asymmetric public-key cryptography uses two keys: a public key and a private key. It is often used for authentication and data encryption. In addition, the hash function outputs a fixed-size value called 'hash' after accepting the content of a QR code as input. The original text cannot easily be recovered by processing the hash result because a hash function is a one-way procedure. Using safe and reliable hash functions, it is impossible to have the same hash value within two QR Codes. Symmetric-key encryption is a system that uses the same secret key for encryption and decryption.

Although symmetric-key algorithms are thought to be simpler and faster than asymmetric ones, the key exchange should be done securely. Public-key cryptography is used in Digital Signatures (DS), a security technique that verifies the content's authenticity, non-repudiation, and data integrity. After computing the hash, it signs the document using the private key. A public-key cryptographic algorithm called Rivest-Shamir-Adleman (RSA) [5] employs a mathematical strategy based on prime numbers. For data encryption and digital signatures, RSA is frequently employed. In contrast, the widely-used public-key cryptographic algorithm for digital signatures is the Elliptic Curve Digital Signature Algorithm (ECDSA). The Advanced Encryption Standard (AES) is a well-known symmetric-key method that is used to encrypt electronic data and is regarded as a very secure algorithm for confidentiality. In [1], the authors have used the lexicography of the URLs to predict if the URL is malicious or not based on the above cryptographic method's presence. But it takes significant time to decode the QR

code which the model proposed in this study is saving. After decoding the URL, the authors are applying Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), K-Nearest Neighbors (K-NN) and Decision Tree (DT) classifiers and evaluating using a 10-fold cross-validation. The authors are obtaining an accuracy of 90.24% using Decision Trees which is the highest among all their other classifiers. The deep learning models used in this paper significantly outperform the achieved accuracy of the classifiers used in [1]. As discussed in [4], QR based security solutions against malware are of high importance in the current situation both in the industry and the science community. The solution proposed in this paper aims to fulfill that need by tackling the problem of increased number of QR related malicious activities using our high performing deep learning classifier.

## 3. Analysis of the malicious QR Code

Quick Response Code (QR Code) is a two dimensional barcode invented in 1994 by an automotive company named Denso Wave to keep a track of its automotive production. It has been widely used across the world since then. Its usage and adoption has especially exploded since the boom of the internet and smartphones in the emerging markets. It has since been very popular in the fintech industry to make fast payments by scanning the receiver's QR code or verifying one's identity using a QR code. As shown in figure 3, QR contains 4 types of cells :

1. The light gray cells represent the fixed pattern in the QR code.
2. The gray cells represent the fixed pattern in the QR code.
3. The black cells represent the format info.
4. The information is contained in the white cells using the following coding: D $\rightarrow$ represents data, E $\rightarrow$ represents Error Correction, X $\rightarrow$ represents Unused Error Correction Level.

A malware QR is generated by implementing a variation of the white cells containing D, E, X and message data in a way that is similar to how a malware URL is implemented using a variation of the lexicographical structure of the benign URL. For this same reason, we can detect patterns in the QR code images and extend this pattern prediction to malicious QR prediction without decoding the QR to its corresponding URL first. This analysis and prediction will hold true since the URL structure maps one-to-one with message data contained in the white cells of the QR block.

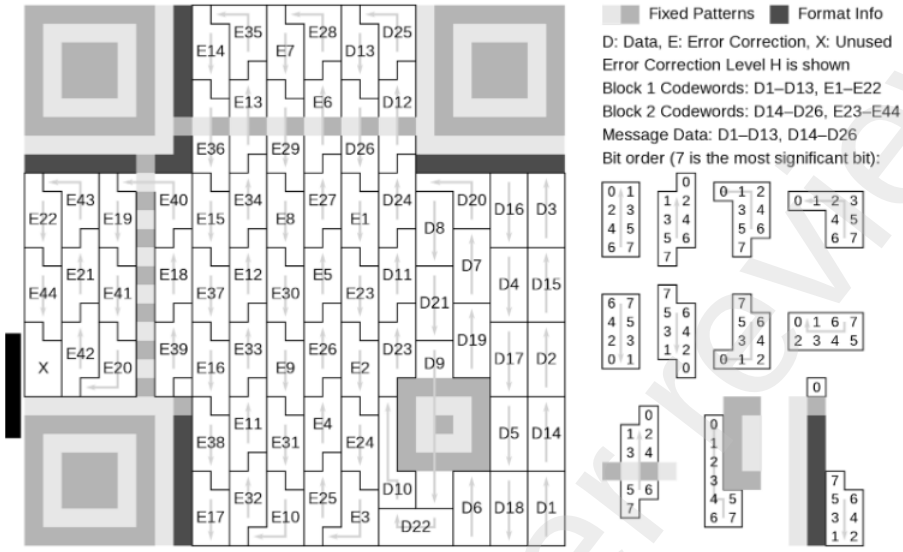



Figure 1: Architecture of Quick Response Code

The methodology described in the next section improves on all previous such studies conducted in the same field since it saves on the decoding time of the QR code and directly shows if the QR is malicious or not. This helps in two major ways:

1. It saves time in case the QR is malicious by not decoding the code further to its corresponding URL.
2. Decoding the malicious QR first exposes the system to the malicious URL, which compromises the system in certain cases of URL malware.

By implementing a front-gate security, a higher security standard has been ensured.

## 4. Methodology

The methodology adopted to build the secure QR system is discussed step by step as follows:

### *4.1. Collecting dataset*

The dataset used for the research is publicly available at [11]


Figure 2: Deep Learning Pipeline

This dataset is created using Python code to generate QR codes from an authentic list of URLs publicly available at [10]

The dataset comprises more than 600,000 URLs. For this study, only the first 100,000 URLs from each class, Benign and Malicious, are used to generate the QR codes. There are 200,000 QR codes images in the dataset in total that encode real-world URLs. Both the Benign QR codes and Malicious QR codes were generated using a single loop of 100,000 iterations each.

The QR Code images that belong to the malicious URLs category are indexed under the 'malicious' folder with 'malicious' as the image file label. Similarly, the QR Codes that belong to the benign URLs category are indexed under the 'benign' folder with 'benign' as the image file label.

It is not recommended that the malicious QR codes are scanned as they contain real malicious URLs that can severely compromise user security and data.

### *4.2. Building and training deep learning model*

Two state-of-the-art CNNs have been used along with Vision Transformer and LinFormer, which is discussed thoroughly in section 6. The model is trained using the collected dataset after doing some important image pre-processing. Once the model is trained, tested and validated against the exhaustive dataset of QR codes, it is used to develop the secure QR System. The deep learning pipeline is shown in figure 2

### *4.3. Developing a secure QR system for filtering malicious QR codes*

The deep learning model developed in this study is hosted in a Flask backend in python using the generated model weights .pth file on the cloud. The classification of QR into malicious or benign can be queried by any 3rd party using this API by making an API call to the model. The companies that use QR Scanner in their app can integrate this model using the API. The architecture of the API is shown in figure 3.

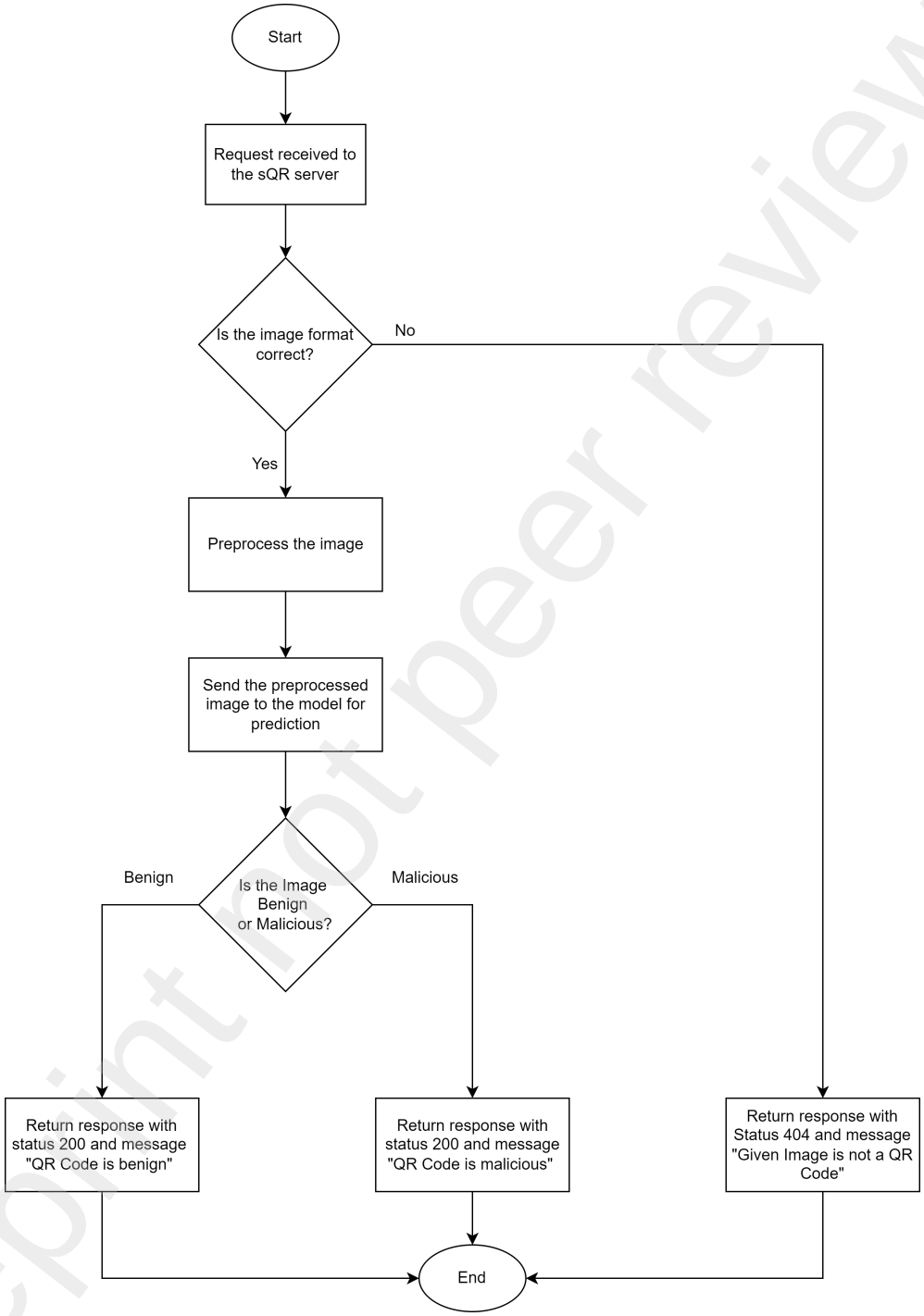



Figure 3: Architecture for the API integrated with the model

## 5. System Information and Datasets Used

- Platform: 4 vCPUS, 16GB RAM.
- GPU Specs: 1 NVIDIA T4 Tensor Core GPU
- Programming Language: Python

## 6. Techniques and Outcome Evaluation

In this section we discuss the outcomes of 3 different deep learning models that are built suitable to the nature of our image dataset.

The first study contains the outcomes of our deep learning model that is built using VGG16 [6]. Firstly, for the study, the generated QR Code images have been resized to 224x224x1 by converting from GBR to Grayscale. Then the image tensor was converted and normalized by keeping the standard deviation and mean of the tensor=0.5. For this research study, the train size of the dataset for the deep learning model is 128,000 images which consists of 64,000 malicious and benign labeled images each. The test size for the same is 32,000 images which consists of 16,000 malicious and benign labeled images each. Finally, the validation size for the deep learning model is taken as 40,000 images which consists of 20,000 benign labeled images and 20,000 malicious labeled images.

The model has been trained with learning rate = 3e-5 and gamma=0.7 (parameter for learning rate scheduler, StepLR) as parameters for only 2 epochs as top class results achieved without the model overfitting in 2 epochs itself.

The image is feeded to the VGG16 which performs certain computations and gives an output. Once the output is received, cross entropy loss is calculated. The loss is then used by the Adam optimizer to calculate new weights and bias for updation. StepLR scheduler is used to decay the learning rate.

In the 2014 ILSVRC competition, VGG-16 was among the architectures with the best performance. With a top-5 classification error of 7.32%, it came in second place behind Google Net, which had a classification error of 6.66%. With a localization error rate of 25.32%, it also won the localization task. The input for VGG16 is a (224, 224, 3) tensor. The first 2 layers use the same padding and channels along with a 3X3 kernel. This is followed by a max pool layer of (2,2) stride, whose output is then feeded into a convolution layer having a filter size of 128, again followed by the same layer. There are

then 256 filters spread across 2 convolution layers with filter sizes of 3 and 3. It is again followed by similar max pool and convolution layers but of different filter sizes. To prevent the spatial characteristic of the image, 1-pixel padding is applied after each convolution layer. A (7, 7, 512) feature map is obtained after the last max pool layer. The output of the final layer is flattened so that it can be feeded to the fully connected layer. The size of the resultant vector is (1,25088). The 3 fully connected layers are followed by a softmax function to classify 2 classes (specific to our problem). The activation function used in each hidden layer is ReLu. ReLu is often the preferred activation function for CNNs.

Cross Entropy Loss [9] has been used as the loss function for all the deep learning models. Cross Entropy is the measure of the difference between two probability distributions for a given random variable or set of events where lower probability events contain more information and higher probability events contain lesser information. Information is represented as a function of probability in equation 1

$$h(x) = -log(P(x)) \tag{1}$$

In terms of entropy, a skewed probability distribution would have low entropy and an equal probability distribution would have a high entropy. Entropy $H$ is represented as a function of probability in equation 2

$$H(X) = -\sum_{x} XP(x) * log(P(x)) \tag{2}$$

Cross-entropy improvises on the idea of entropy and calculates the number of bits required to represent or transmit an average event from one distribution compared to another distribution. The equations for cross entropy of events A and B whose probability is shown in equation 3

$$H(P, Q) = -\sum_{x} XP(x) * log(Q(x)) \tag{3}$$

where P(x) is the probability of the event x in P, Q(x) is the probability of event x in Q and log is the base-2 logarithm, meaning that the results are in bits.

Adam Optimiser [3] has been used for optimizing the objective function of all the deep learning models. Adam or Adaptive Moment Estimation is

an improvisation combining the strengths of RMSProp optimiser and AdaGrad optimiser. In this optimization algorithm, running averages of both the gradients and the second moments of the gradients are used. StepLR Scheduler has been used in the deep learning model for learning rate scheduling to further optimize the learning rate performance.

The metrics used for judging the model are:

1. Accuracy, as shown in equation 4
2. Precision, as shown in equation 5
3. Recall, as shown in equation 6
4. F1 Score, as shown in equation 7

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7}$$

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

Another important metric used to analyse the results is FPR, as shown in equation 8

VGG16 Results: The model gives an excellent outcome by rendering an accuracy of 99.91% on train data and 99.95% on test data. The model is trained for 2 epochs as it was deemed sufficient based on the results.

This section contains the outcomes of our deep learning model that is built using EfficientNet Convolutional Neural Network [7].

The same preprocessed dataset is fed into the EfficientNet model with the same train, test and validation data sizes. The model has been trained with lr=3e-5 and gamma=0.7 as parameters for 2 epochs as EfficientNet also gave similar results as Vision Transformer.

In order to scale up CNNs in a more organized way, EfficientNet suggests a novel model scaling technique that makes use of a straightforward

but incredibly powerful compound coefficient. EfficientNet uniformly scales each dimension with a fixed set of scaling coefficients, in contrast to existing approaches that scale network dimensions like width, depth, and resolution. EfficientNets superpass state-of-the-art accuracy with up to 10x greater efficiency leveraging innovative scaling strategy and recent advancements in AutoML.

Cross Entropy Loss has been used as the loss function for this deep learning model along with Adam optimizer to optimize the objective function and StepLR learning rate scheduler to decay the learning rate.

EfficientNet Results: The model gives a good outcome by rendering an accuracy of 99.84% on train data and 99.91% on test data. The model is trained for 2 epochs as it was deemed sufficient based on the results.

This section contains the outcomes of our deep learning model that is built using Vision Transformer [2].

The same preprocessed dataset is fed into the Vision Transformer model with the same train, test and validation data sizes. The model has been trained with lr=3e-5 and gamma=0.7 as parameters for 2 epochs as Vision Transformer also returned similar values as EfficientNet. For this study LinFormer [8] has been used along with Vision Transformer. LinFormer is a Linear-Time Transformer Architecture introduced by researchers of Meta AI. It is the first theoretically proven linear-time transformer architecture. It is both memory and time efficient. It is implemented in order to reduce the cost of training and deploying our large transformer model. Since the key efficiency bottleneck in our transformer model is encountered in the model's self-attention mechanism, using the LinFormer architecture brings down the overall self-attention complexity from $O(n^2)$ to $O(n)$ in both time and space complexity by using a low-rank matrix instead of attending to all the tokens at each layer where the representation of each token is updated by visiting the tokens present in the previous layer.

Vision Transformer (ViT) is a transformer architecture introduced by the Brain Team, Google Research, that is targeted at vision processing tasks like image processing. Attention is a quadratic task and therefore, the amount of computation should be huge for images. The images can be of big sizes, and we have to make sure that each pixel attends to each pixel. Some solutions were introduced for doing local attention in images like convolutions, where a kernel is used to slide along in the image using a sliding window. Vision Transformer suggests that we do global attention. So suppose if a 224x224

Table 1: Training Results along with Confusion Matrix

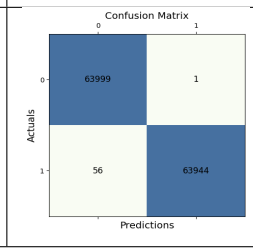
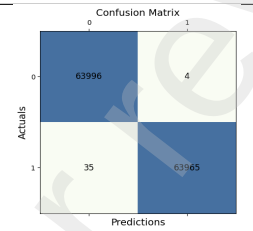
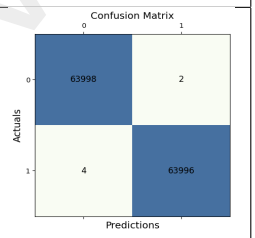
| Train Results | | | |
|---|---|---|---|
| Metrics | VGG16 | EfficientNet | ViT |
| Accuracy | 99.9100% | 99.8400% | 99.9600% |
| Precision | 99.9131% | 99.8475% | 99.9568% |
| Recall | 99.9078% | 99.8421% | 99.9554% |
| F1 Score | 99.9074% | 99.8421% | 99.9554% |
| Confusion Matrix |  |  |  |

image is taken, it will be divided into 14x14 or 16x16 patches. These patches will then be flattened in the sequence 14x14 $\rightarrow$ 196x1 or 16x16 $\rightarrow$ 256x1. In the base version of the model, they get a 796x1 matrix after flattening the patch. A position encoding of the image patches is performed and passed to the transformer encoder which is the same as the one used in original transformers. The output of the transformer encoder is finally passed on to the Multi Layered Perceptron Head (MLP Head) which classifies the image.

This deep learning model's loss function is Cross Entropy Loss, and the objective function is optimised using the Adam Optimizer. The decay of the learning rate is managed by the StepLR learning rate scheduler.

Vision Transformer Results: The model gives an excellent outcome by rendering an accuracy of 99.96% on train data and 99.965% on test data. The model is trained for 2 epochs as it was deemed sufficient based on the results.

The results of all the models for training, validation and testing are tabulated in Table 1, Table 2 and Table 3

In the Confusion Matrix, 0 indicates 'benign' and 1 indicates 'malicious'.

00: True Negatives

01: False Positives

10: False Negatives

11: True Positives

Table 2: Validation Results along with Confusion Matrix

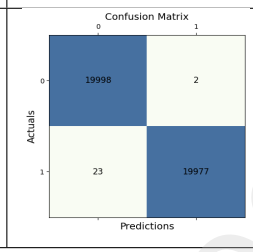
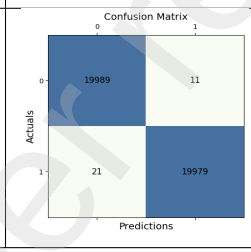
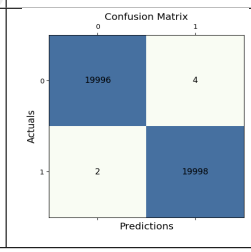
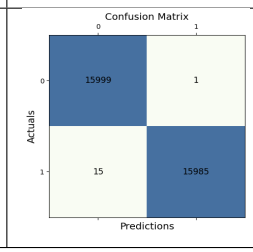
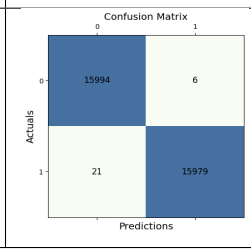
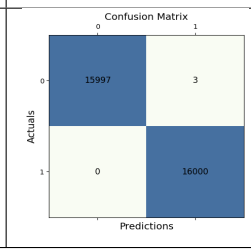
| Validation Results | | | |
|---|---|---|---|
| Metrics | VGG16 | EfficientNet | ViT |
| Accuracy | 99.9382% | 99.9207% | 99.9857% |
| Precision | 99.9396% | 99.9224% | 99.9665% |
| Recall | 99.9375% | 99.9200% | 99.9656% |
| F1 Score | 99.9375% | 99.9199% | 99.9656% |
| Confusion Matrix |  |  |  |

Table 3: Test Results along with Confusion Matrix

| Test Results | | | |
|---|---|---|---|
| Metrics | VGG16 | EfficientNet | ViT |
| Accuracy | 99.9500% | 99.9156% | 99.9656% |
| Precision | 99.9514% | 99.9185% | 99.9665% |
| Recall | 99.9500% | 99.99156% | 99.9656% |
| F1 Score | 99.9499% | 99.9157% | 99.9656% |
| Confusion Matrix |  |  |  |

## 7. Discussion

It is observed that in all the 3 deep learning models, the accuracy is around 99.9% which is considered exceptional. This is in line with our expectation since the QR code was built in order for users to be able to scan and decode the underlying URL very quickly. For this same reason, the QR code contains repeating patterns that are picked up and learnt by neural networks very well. For this research study, accuracy has been chosen as the main metric as the dataset doesn't have any class imbalance. Based on the accuracy, Vision Transformer outperformed the EfficientNet CNN by 0.05% and hence, the Vision Transformer based deep learning model is chosen for solving the malicious QR classification problem. Classifying malicious QR code with an accuracy above 99.96% and precision of above 99.96% ensures that the secure QR code system will be highly efficient in filtering out malignant QR codes corrupted by hackers from benign QR codes. This system has been trained on a very large and authentic dataset and thus ensures that the system will perform very well in real world scenarios. This model can be deployed to classify QR codes in real-time as a user interacts with the QR code. Since the model directly classifies the QR based on the image and doesn't convert it to its corresponding URL, it saves significant time in the decoding process and combined with the high accuracy and efficiency of the proposed secure QR system is unmatched and will provide significant value to the current industry needs.

## 8. Conclusion and Future Scope

Since the model is very lightweight and works in near real time, this can be integrated in the current QR systems to enhance it's security, especially in the domains where malicious QR activity is significantly prevalent like India's United Payments Interface (UPI) QR codes, QRs involved in the KYC processes, Identity Management like event registration, while looking up food menus through QR, Google Lens, and other QR scanners. The model can be retrained with newer QR image dataset periodically in order to ensure the filtering system is up to date with the latest pattern of malicious QR code. The architecture of the model would still remain the same. Wide Scale adoption of the secure QR system would provide a great impact in bringing down malicious activities using QR codes.

## References


[1] Mohammed S Al-Zahrani, Heider AM Wahsheh, and Fawaz W Alsaade. Secure real-time artificial intelligence system against malicious qr code links. *Security and Communication Networks*, 2021, 2021.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[4] Katharina Krombholz, Peter Frühwirt, Peter Kieseberg, Ioannis Kapsalis, Markus Huber, and Edgar Weippl. Qr code security: A survey of attacks and challenges for usable security. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, pages 79–90. Springer, 2014.

[5] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[8] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[9] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[10] Benign and Malicious URLs Kaggle Dataset

[11] Benign and Malicious QR Codes Kaggle Dataset