Baghdad Science Journal

Volume 23 | Issue 1 Article 27

1-14-2026

Robust QR Code Counterfeit Detection by Integrating VGG19 Neural Embedding and Triplet Loss

Gede Putra Kusuma
Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480, inegara@binus.edu

Yulianto Yulianto
Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480, yulianto003@binus.ac.id

Renaldy Fredyan
Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480, renaldy.fredyan@binus.ac.id

Hendi Chandi
Advotics Teknologi Global, Jakarta, Indonesia 12950, hendi.chandi@advotics.com

Jeffry William Tan
Advotics Teknologi Global, Jakarta, Indonesia 12950, jeffry.tani@advotics.com

See next page for additional authors

How to Cite this Article

Putra Kusuma, Gede; Yulianto, Yulianto; Fredyan, Renaldy; Chandi, Hendi; William Tan, Jeffry; Kwan, Philip; and Dafa Syukur, Muhammad (2026) "Robust QR Code Counterfeit Detection by Integrating VGG19 Neural Embedding and Triplet Loss," *Baghdad Science Journal*: Vol. 23: Iss. 1, Article 27.
DOI: https://doi.org/10.21123/2411-7986.5189

Robust QR Code Counterfeit Detection by Integrating VGG19 Neural Embedding and Triplet Loss

Authors

Gede Putra Kusuma, Yulianto Yulianto, Renaldy Fredyan, Hendi Chandi, Jeffry William Tan, Philip Kwan, and Muhammad Dafa Syukur

BAGHDAD SCIENCE JOURNAL 2026;23(1):356–369

RESEARCH ARTICLE

Robust QR Code Counterfeit Detection by Integrating VGG19 Neural Embedding and Triplet Loss

Gede Putra Kusuma [1], Yulianto [2], Renaldy Fredyan [2,*], Hendi Chandi [3], Jeffry William Tan [3], Philip Kwan [3], Muhammad Dafa Syukur [3]

[1] Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480
[2] Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480
[3] Advotics Teknologi Global, Jakarta, Indonesia 12950

ABSTRACT

Detecting counterfeit Quick Response codes plays a critical role in protecting the integrity of products and documents. This paper presents a novel methodology to increase the reliability of QR code counterfeit detection by integrating neural embedding VGG19 and triplet loss. The approach uses deep neural networks. Specifically, the modification of the VGG19 architecture is used to extract unique features from QR codes. These features are embedded in a multi-dimensional space using neural embedding methods. The use of the triplet loss function aims to increase the discriminative power of the embeddings, ensuring the discriminative power of the embeddings, and ensuring that authentic QR code embeddings are closer to each other and further away from counterfeit embeddings. The effectiveness and robustness of the proposed methodology is substantiated by extensive experimentation on a large dataset. The proposed system effectively identifies counterfeit codes with high confidence, with validation and test scores of up to 99.87% and 99.55% respectively, even when images are distorted, cropped and noisy. Practical implications of the research will be explored, highlighting possible applications for product authentication, anti-counterfeit strategy, and document verification. The combination of neural embedding VGG19 and triplet loss results in improved detection accuracy, enhancing the security and reliability of QR code-based systems. This approach offers a strong answer to the increasingly difficult problems of counterfeit detection in the modern digital age, therefore reflecting a major progress in the area of security and authentication technology. It improves the dependability and integrity of digital systems in addition to solving the technological challenges related to information and counterfeit product identification.

Keywords: Counterfeit detection, Neural embedding, QR code detection, Triplet Loss, VGG19 model

Introduction

Two-dimensional (2D) matrix coding can be used to efficiently store machine-readable data. [1] Widespread use of mobile phones has made 2D matrix codes useful in many different areas of personal and professional life. [2] The QR (Quick Response) code was originally developed by Denso Wave for the Japanese automotive industry in 1994. [3] Since then, however, its use has been extended to cover a much wider range of applications. They are widely used in industries such as manufacturing, logistics, distribution, media, advertising, tourism, e-commerce, identity and authentication. [4] The QR code often contains additional

Received 19 March 2024; revised 3 February 2025; accepted 5 February 2025.
Available online 14 January 2026

* Corresponding author.
E-mail addresses: inegara@binus.edu (G. P. Kusuma), yulianto003@binus.ac.id (Yulianto), renaldy.fredyan@binus.ac.id (R. Fredyan), hendi.chandi@advotics.com (H. Chandi), jeffry.tani@advotics.com (J. W. Tan), philip.kwan@advotics.com (P. Kwan), dafa.syukur@advotics.com (M. D. Syukur).

https://doi.org/10.21123/2411-7986.5189

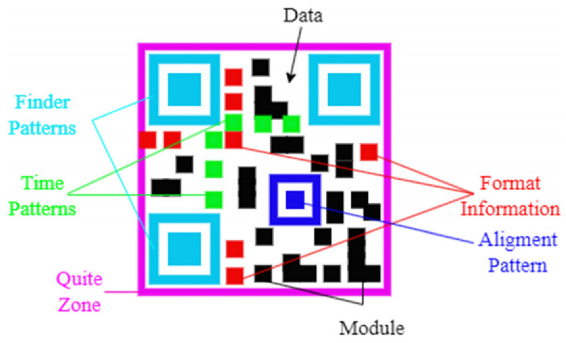


Fig. 1. Quick response (QR) Code, version 2, 25 modules by 25 modules.

information about the item, object or location.[5] The QR code information on the above[6] may also include a website Uniform Resource Locator (URL), global Positioning System (GPS) coordinates, contact details, a shipping destination, payment information, and so on.[7] The QR code model was originally developed to a group of two-dimensional matrix codes.[8] It includes QR codes and has similarities to data matrix codes. The conventional QR codes, namely QR code models 1 and 2, have a square shape and have distinctive square patterns, known as Finder Patterns (FP), located at each of the three corners of the square.[8] To identify the QR code and determine its exact position, size, and orientation, the above patterns are used.[8] The composition of a QR code can be described in the following way. See Fig. 1.

The research problem related to the QR code identification is that QR code images not only provide flexibility to share information about anything based on image visual recognition but also can provide secret information for individuals.[9] For example, when the customer candidate books a train and plane ticket online, the customer receives a one-page information sheet that includes a QR code image to show the fastest way to check in when the customer arrives at the airport.[10] QR code generated by the system and intended only for individual users, if another person who has copied QR code without permission, whether by copying with a photo or by printing the QR code image, they may have chance to invade place where QR code is needed to authenticate. Those issues known as industrial counterfeiting.[11] To prevent the QR code image from being counterfeited, the following points needs to be considered.[12]

Over the last few years, companies have sought to prevent product and QR code counterfeits during supply chain tracking by considering the loss threshold between the original pattern embedded into the QR code and the counterfeit QR code.[13] Blockchain architecture is being implemented to prevent fakes, goods and QR codes by controlling every process of transition.[14,15] The issue of counterfeit detection has become a prominent concern in contemporary society the issue of detecting fakes has become a prominent concern in modern society due to the proliferation of fake goods and fraud, which pose significant risks to consumer welfare, brand reputation and overall economic security. QR codes, widely used to authenticate products[16] and retrieve information,[17] have been shown to be highly susceptible to counterfeiters. As a result, there has been a growing demand for robust and efficient techniques for the identification of counterfeit QR codes and the assurance of their integrity.

This paper presents a novel method to effectively detect counterfeit QR codes by integrating neural embedding VGG19[18] and triplet loss.[19] Deep learning models are well suited to analyze and represent complex data, including QR codes, as they have shown exceptional ability to encode textual and visual information. It is possible to extract discriminative features from QR codes using the VGG19 architecture, a widely recognized and effective image classification model. These features can be used for capturing unique patterns and characteristics inherent in code. The main innovation of this approach is the use of neural embedding techniques for the transformation of the extracted features into a high-dimensional space. The process of embedding enhances the discriminatory power of the features and makes it possible to clearly distinguish between authentic QR codes and fraudulent ones. The use of the triplet loss function is implemented to improve the quality of the embeddings. This loss function is used to create greater separation between fake QR code embeddings, while also making genuine QR code embeddings more closely spaced. Extensive experimentation on a large dataset has shown that the proposed methodology has excellent performance in counterfeit detection. The system is resilient to many of the obstacles commonly encountered in real-world situations. These include image distortion, cropping and noise intrusion. The practical implications of this research are noteworthy, as the identification of fraudulent QR codes can play a crucial role in product authentication, anti-counterfeiting strategies, and document verification. Previous studies have demonstrated the effectiveness of leveraging trustworthy AI techniques to enhance intrusion detection systems, such as those found in the work by Aljanabi,[20] which can provide valuable insights and strategies applicable to the challenges addressed in this study.

The task of neural embedding is to identify class names from given sample data, based on the convolutional neural network of the deep learning architecture. The complex process behind deep learning attempts to transform the images of the sample

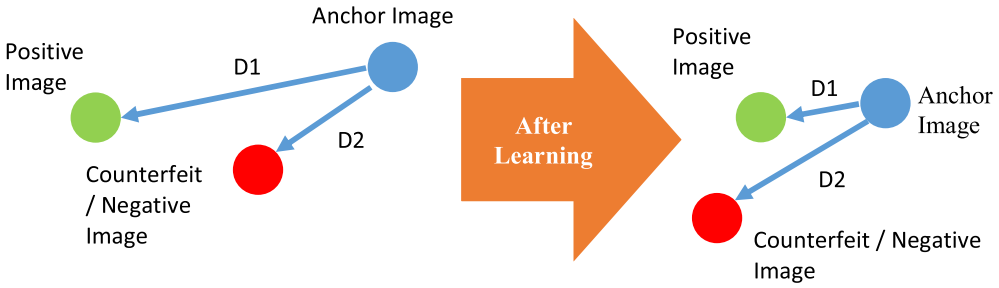


Fig. 2. The relationship between the anchor images with the positive and counterfeit images in the dimension of the feature vector space. The objective is to minimize the intra-class distance (D1) between anchors with positive images and maximize the inter-class distance (D2) between anchors with false images.

data so that they become data of a different dimension in the feature space. As shown in Fig. 2 which illustrates the feature vector of the 3 samples, which is driven by the output of the neural embedding given 3 sample images. The sample is a circle of red, circle of green, and blue. The red circle represents a negative sample or a fake sample or a counterfeit sample. The blue circle is a sample anchor that will serve as the main reference point, and the green circle is the positive sample or the original sample. The task of the neural embedding architecture is to minimize the intra-class distance (D1) between the positive sample and the anchor sample after training (learning) on a large dataset and increase the distance between the classes (D2) between the negative sample and the anchor sample. In this study, a novel framework architecture is proposed that can be used for intra-class distance minimizing (D1) and interclass distance enlargement (D2). Inspired by the Siamese triplet loss network, which uses triplet images to train its architecture, in this study, the backbone architecture is modified by using a modified VGG19.

The innovative nature of this study is illustrated by the following elements:

a. VGG19 deep learning as the backbone architecture is used to identify whether the input image of the QR code is an original or a counterfeit.
b. The VGG19 architecture requires an input layer to be in a 3-dimensional form, so additional layers of a 2-dimensional convolutional neural network and an activation function are paired with the input to adapt the difference of the dimensional gap between the input image with a single layer to 3 layers.
c. The triplet loss training technique proposed to improve the classification accuracy of VGG19 with the learning objective to maximize an interclass loss value of two group class anchor images with counterfeit image and minimize the intraclass loss of group images between an anchor and positive image.
d. The downstream architecture is also proposed to improve the accuracy of the VGG19.

Materials and methods

The Visual Geometry Group (VGG19) architecture was designed for image classification, which can be used to identify the image object with a multi-label.[21] But never rule out the possibility of making changes. In this study, the VGG19 architecture was modified. By using the Pytorch library the VGG19 architecture can be easily used without having to build the architecture from scratch.[22] In principle, the VGG19 architecture consists of a block for the extraction or embedding of features and a block for the classification using dense network. See Fig. 3.

The novelty of this paper is the two-stage of the learning process that can be used to identify whether a QR code image input to the proposed method is genuine or counterfeit. In the first stage, a learning algorithm for feature embedding is proposed using the architecture shown in Fig. 4. The feature embedding learning acts as an upstream model of

Fig. 3. VGG19 architecture.

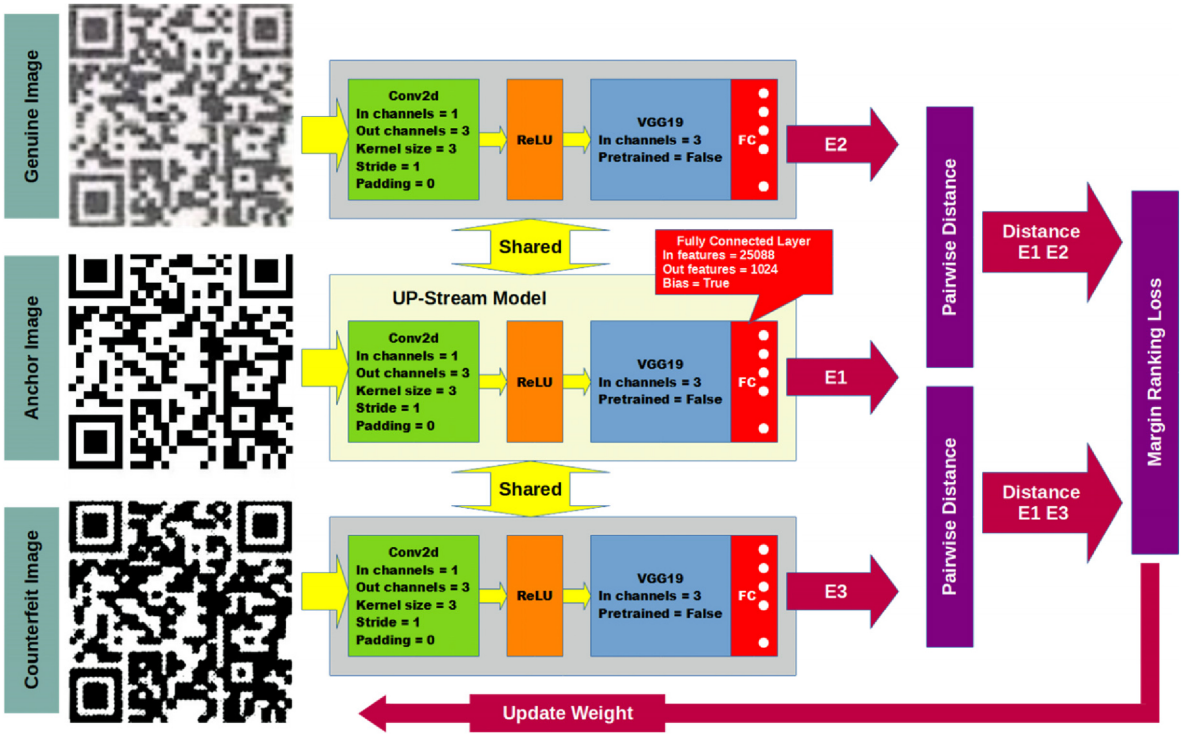


Fig. 4. Modification of the VGG19 architecture in the triplet loss scheme as an upstream model for feature embedding learning on this study.

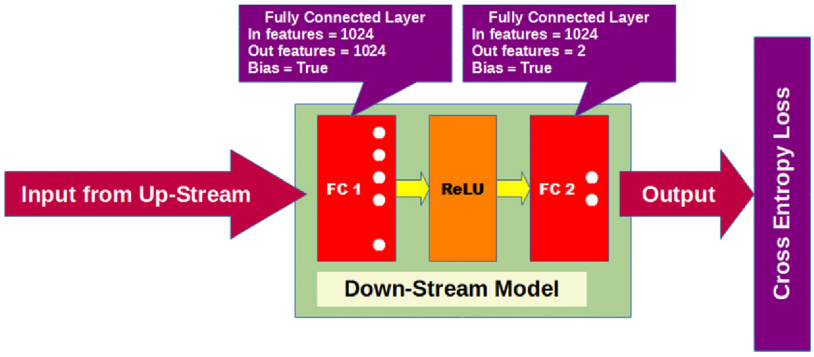


Fig. 5. The downstream model was used as the classifier model proposed in this study.

the architecture. In addition, for the second stage, the goal is to combine the pre-training model of embedding learning in Fig. 4 with the downstream framework, which consists of the neural classifier framework shown in Fig. 5. In the VGG19 architecture, after the last pooling layer, namely the adaptive average pooling layer, the output dimension is 7×7 pixels with a total number of layers of 512. The output adaptive average pooling layer is then flattened into a 1D array with 25088 neurons. The classification block, consisting of dense neural networks, is then connected to the 1-dimensional array output, which represents a feature vector. In this study, the workflow of the proposed method can be seen in Fig. 4 that illustrates the modification of the VGG19 architecture in the triplet loss scheme, which serves as the upstream model for feature embedding learning. This modification enhances the ability of the model to discriminate between authentic and counterfeit QR codes by optimizing the embedding space.

For the first change, the classification block was removed. As a substitute, the custom of a Fully Connected Layer (FC) was added. The FC layer used has the following parameters; the input feature for the FC layer is 25088 neurons, and the feature output is set to 1024 neurons, with the bias parameter set to True. By default, VGG19 architecture requires image input dimensions of 224 $\times$ 224 pixels, by the number of layers is 3, because image input is in Red, Green, Blue (RGB) form. On the other hand, the input image used

in this experiment is in a grey scale format, which has only one layer of color. The second modification was then added to this study to bridge the incompatibilities between the greyscale image dimension and the input requirements of the VGG19 architecture. Before entering the VGG19 architecture, a single convolution layer and an activation function were added. The convolution layer added has the following parameters; the convolution layer used is two-dimensional with the input channels set to 1, and the kernel size used for the matrix convolution operation is 3×3. The step parameter on the added convolution layer is set to 1 step. The addition padding for the input image is set to 0 or no edge addition. And the output channels on the convolution layer addition are set to 3 layers. An activation layer must be added after the convolution layer to prevent accuracy performance from degrading. The ReLU activation function was chosen to be placed after the addition of the convolution layer. The details of the layer addition for convolution layer and activation function can be seen in Fig. 4.

As mentioned above, the neural embedding framework proposed in Fig. 4 needs to be trained first. On each forward bias in training mode, the upstream model is used three times to compute feature vectors, so this flow is referred to as the shared upstream model process. In the first flow, the upstream model provides the image with the anchor label to obtain the embedding feature, this is generated by the fully connected layer. The embedding output feature, collected from anchor images input, is initialized as E1. For the second model is used again to calculate the vector feature by inserting the real image. The embedding feature resulting from the real image that is inserted is labelled E2. For the third flow in the forward bias, the model is reused by inserting the counterfeit or fake image into the upstream model and at the end output the E3 is generated. The embedding of the feature vector that is generated from the FC layer and is denoted as E1, E2, and E3 for each notation, has a dimension length of 1024.

$$L_{tl} = \frac{1}{N}\sum_{i}^{N} max\left[d\left(z_a,\ z_p\right) - d\left(z_a,\ z_n\right) - m,\ 0\right], \quad (1)$$

The applicability of the triplet loss formula inspired by the research proposed by Yan et al.[23] and Boutros et al.[24] The triplet loss is responsible for making the interclass distance or measure between two or more images that have different labels or identities greater than the intraclass distance. To train the deep learning model with triplet loss, three pairs of sample images are required, the first being the image assumed to be the positive label, which may contain some noise, the second being the image assumed to be the negative label or forgery, and the last or third being the clear image without noise declared as the anchor label. The group images that contain sample image anchors marked as x_p, the group of images with the negative label as x_n, and the group of images with the anchor label denote as x_a.

The L_{tl} represents a triplet loss function, where the lower the loss value is close to zero during the training process, the more reliable the model algorithm becomes. The z notation is a feature vector that is generated from the block algorithm of the feature extractor, where in this study the feature extractor is generated using VGG19. The notation of z_a is a feature vector that is derived from an image with an anchor's label that is extracted using a VGG19 architecture. Likewise, the notation z_p is a vector feature generated from an image from a gallery with a positive label, and the notation z_n is a feature vector from an image with a negative label extracted using VGG19. The sub-notation of $d(z_a,\ z_p)$ from Eq. (1) represents the distance measurement between the feature vector of the anchor image (z_a) and the feature vector of the positive image (z_p). The sub-notation of $d(z_a,\ z_n)$ in Eq. (1) is represented as a distance measurement between the feature extraction of anchor image (z_a) and the feature extraction of the negative image (z_n). In this study, the pairwise distance is used to obtain the distance value of $d(z_a,\ z_p)$ and $d(z_a,\ z_n)$. The notation of m represents as the margin constant with a value of 1.0 by default initialization. The notation N indicates how many samples are contained in the minibatch. Every last calculation in forward mode for all the images fed into VGG19 is divided by N, then the backward operation or back-propagation calculation is performed.

$$d\left(z_a, z_x\right) = \|z_a - z_x + \varepsilon e\|_p, \quad (2)$$

$$loss\left(d_1, d_2, y\right) = \max\left(0,\ -y*\left(d_1 - d_2\right) + margin\right), \quad (3)$$

Using Eq. (2), the pairwise distance between the real anchor (E1 paired with E2) and the fake anchor (E1 paired with E3) is counted, with the aim of minimizing the intra-class distance between E1 and E2 and maximizing the inter-class distance between E1 and E3. The output of the pairwise distance calculation is a single scalar loss value that will be used for the margin ranking loss count, can be seen in Eq. (3). Finally, the upstream model is counted backwards using the scalar value of the margin ranking loss.

A shown in Eq. (3) is a criterion that acts as a loss function that is used to measure the rank relationship

between the distance of anchor-positive $d_1(z_a, z_p)$ with anchor-negative $d_2(z_a, z_n)$.[25] The y parameter in Eq. (3) is set to –1.

The downstream model that was also proposed in this study is describe in Fig. 5. The proposed downstream model consists of a combination of two fully connected layers, namely fully connected layer 1 (FC 1) and fully connected layer 2 (FC 2). The connection between FC 1 and FC 2 in the middle added ReLU's as activating functional layers. The output from the upstream model is a vector feature that has been flattened to a tensor array in 1 dimensional form, with the number of features being 1024 vector features. The one dimension of the tensorial array, representing a feature vector, is obtained from the embed process, which is subsequently incorporated into the FC 1 layer. The number of input features for the FC 1 layer is 1024. The output number of the FC 1 layer is also equal to 1024 neurons. The output of FC 1 is then connected to ReLU. From the ReLU output, the computation is passed to FC 2, which has 1024 neural inputs and 2 neural outputs. The reason why the final output of FC2 is provided with only two neurons output is because the case of this study is to identify whether the QR code image fed into the framework is real or fake. The bias parameters on the FC 1 and FC 2 layers are set to true. The loss function used at the very end is the cross-entropy loss.

Results and discussion

While the upstream process training was in progress, the validation process was also in progress after each epoch of operation. The batch size for training the dataset loader was set to 4 and for validating the dataset loader set to 1. The measuring method used to find out how the training data learned by the upstream model is affected by the margin ranking loss. See Eq. (3). The objective function of the Margin Ranking Loss is that the lower the losses (near zero), the better the upstream model learned and the global optimality. Fig. 6 shows the margin ranking loss score from the upstream model.

As shown in Fig. 6, the loss value between the train and the validation is slightly different. The horizontal axis of the line indicates how many epochs there are, starting with the first epoch and ending with 20. The details of the data labels for each point in Fig. 6 are shown in Table 1, for a note, the data and graphic that are presented in Table 1 are only an example of the number of epochs from 1 to 20 epochs are because the 21st epoch to 100 epochs are jammed on the same value. In the first epoch, the training loss was very high, it was 1.083. The validation loss was also at a high level, at 1.071. The VGG19 architecture in the state of learning the QR code images for the first time caused higher loss in the first epoch. The experiment also found that for epochs 19 to 100, the margin ranking loss for the training dataset can be constantly reduced to 0, and by using the validation dataset, the lower loss is constant to 0.019.

At the end of the last 100 epochs, the weight of the pre-trained upstream model is stored. The pairwise distance can also be measured to see the correlation distance using the pre-training of the upstream model. The result of the neural embedding from the

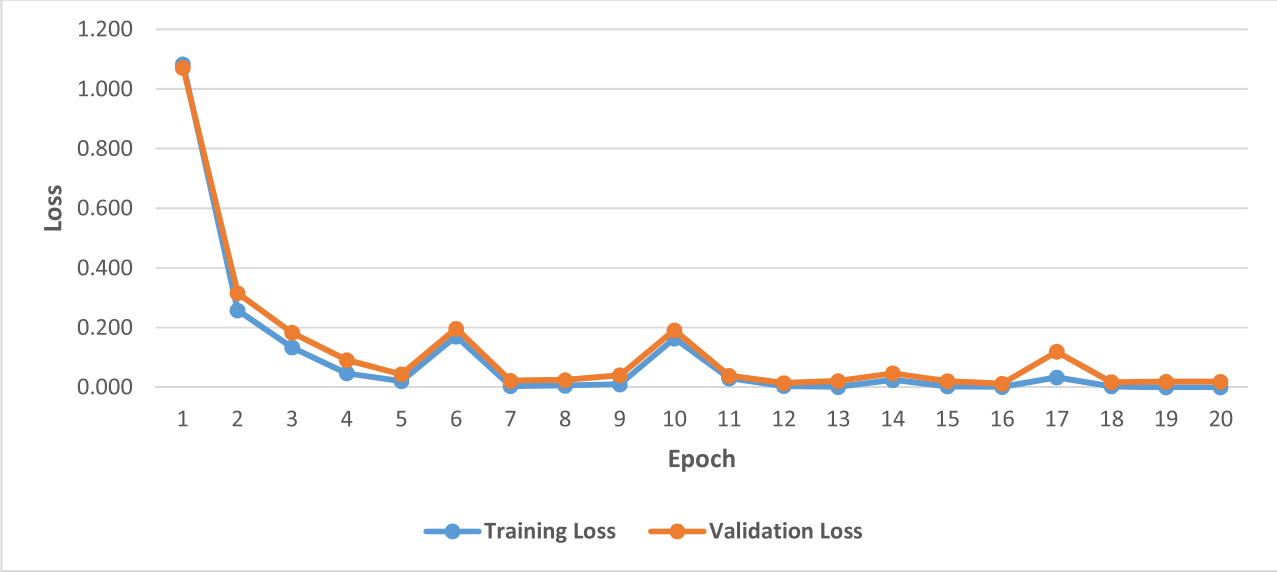


Fig. 6. Training and validation loss measured during the training process.

Table 1. The sample loss state of the upstream model during the training process.

Epoch	Train Loss	Validation Loss
1	1,083	1,071
2	0,258	0,316
3	0,134	0,183
4	0,047	0,092
5	0,021	0,044
6	0,170	0,197
7	0,004	0,022
8	0,006	0,025
9	0,010	0,040
10	0,164	0,191
11	0,030	0,039
12	0,004	0,015
13	0,001	0,021
14	0,024	0,047
15	0,003	0,021
16	0,001	0,013
17	0,033	0,119
18	0,003	0,017
19	0	0,019
20	0	0,019
21–100	0	0,019

pre-train of the upstream model which was used to extract the positive and negative input images as feature vectors, which were then measured using pairwise distance. The point with the blue colour indicates the distance between the anchor image and the positive image, obtained by a distance that is smaller than that of a point with the orange color. Number of 3.706 pairs of anchor and positive images from the training dataset, measured with the pairwise distance $d(z_a, z_p)$ and displayed in blue color, See Fig. 7. Also, the number of 2.180 anchor pairs with negative or counterfeit images from the training dataset, measured by pairwise distance $d(z_a, z_n)$, shown in orange color. Another experiment was also performed on the validation dataset to plot and visualize the mapping position using pairwise distance. The set of 763 image pairs from the validation dataset was extracted using a pre-trained upstream model, then the feature embedding output was calculated using pairwise distance and shown in Fig. 8. The 763 pairs of images from the test dataset were also mapped according to the pairwise distance and are shown in Fig. 9.

After the training process on the upstream model has been completed, with the training and validation scores dropping close to zero, the pre-trained upstream model is then reloaded. A downstream model proposed in Fig. 5 is then merged with the output neural embedding from the upstream model with the number of 1024 features. The training mode that was implemented to merge the two architectures used a classifying mode. The criterion used to update from the downstream model to the upstream model was a cross-entropy loss, with the overall classification required by the merging of two models being two classes. There are positive classes and negative or counterfeit classes. Each of the epochs of the training circle was also followed by the process of validation. To directly monitor whether the loss value can be reduced, the training and validation also measured. Monitoring the loss value also means knowing whether the model can learn the pattern from the data, or whether there is a chance of overfitting or underfitting. The training and validation loss scores for the combined models are shown in Fig. 10. The horizontal axis represents the number of epochs, and the vertical axis represents the value of the loss. In

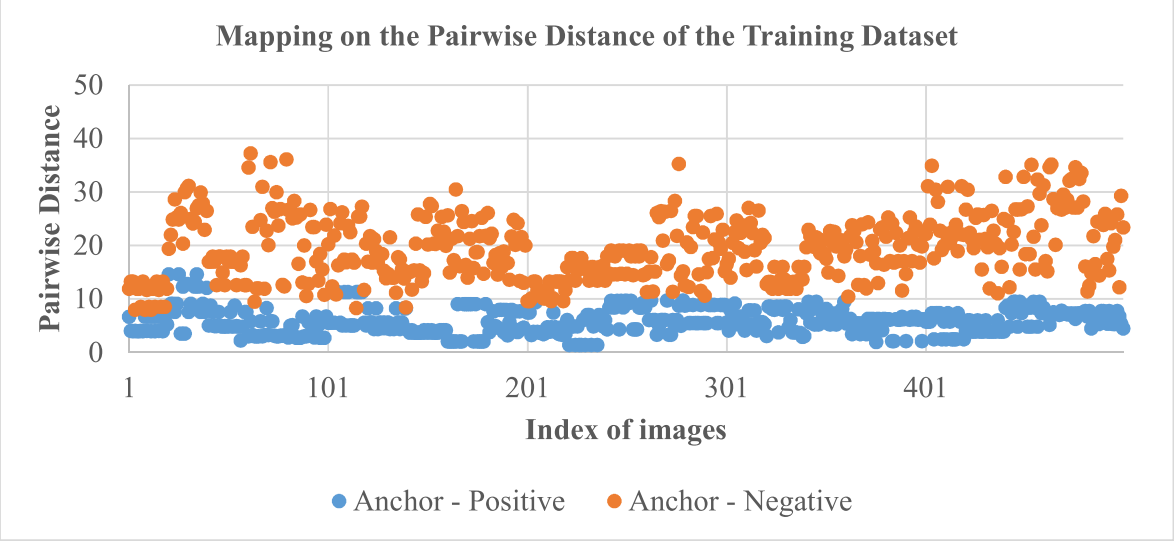


Fig. 7. Training data measured by pairwise distance.

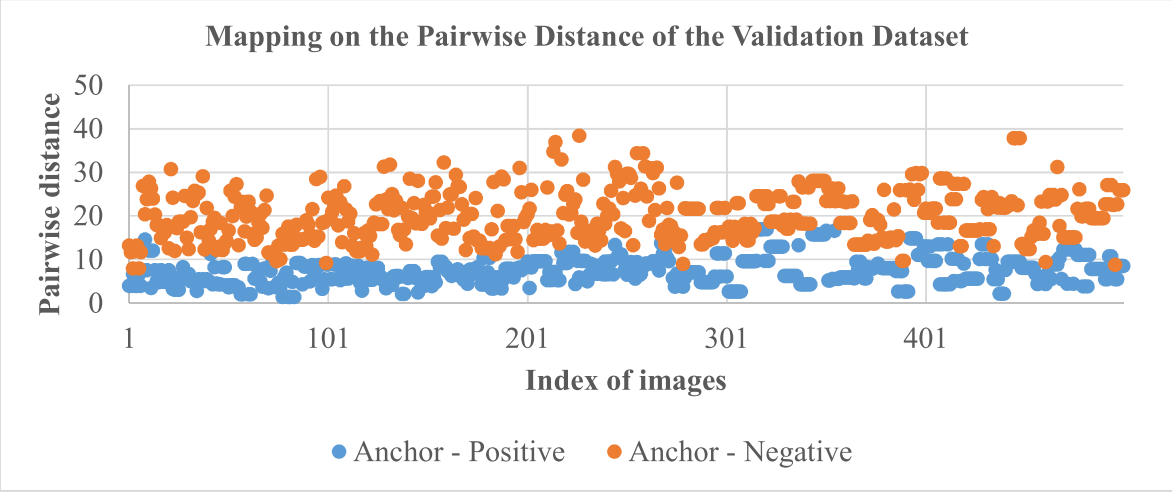


Fig. 8. Validation data measured by pairwise distance.

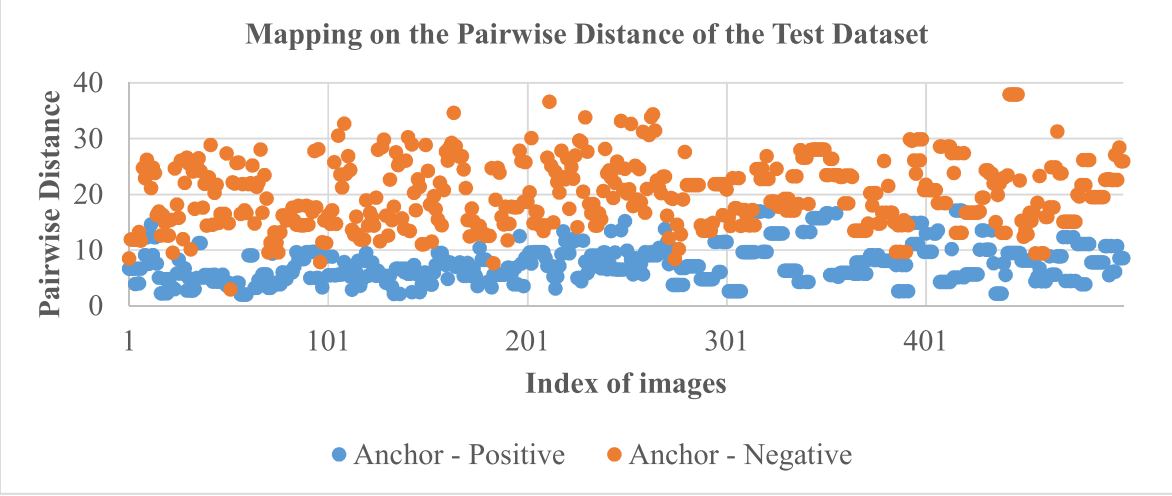


Fig. 9. Test data measured by pairwise distance.

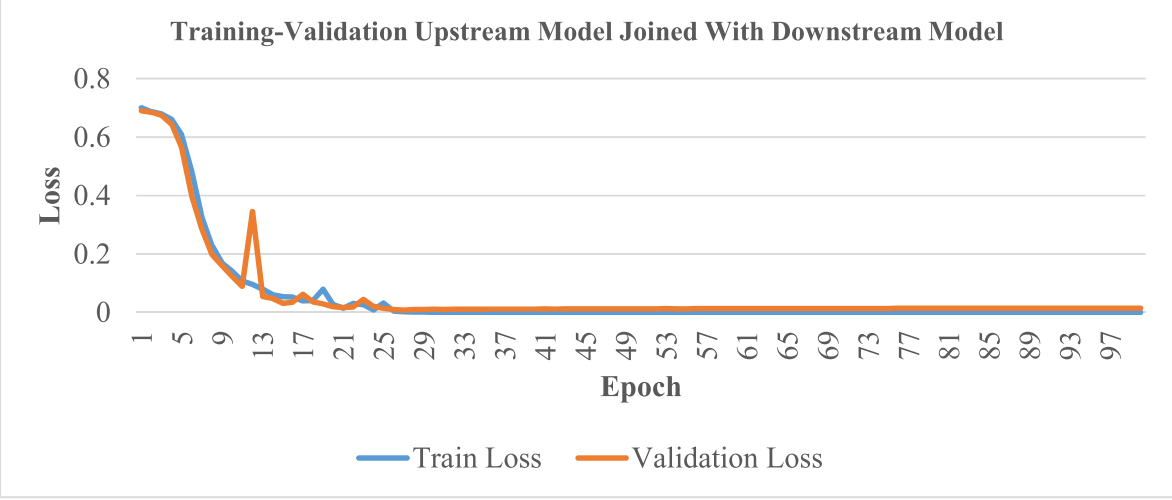


Fig. 10. The measurement of triplet loss during training was performed using the training and validation datasets.

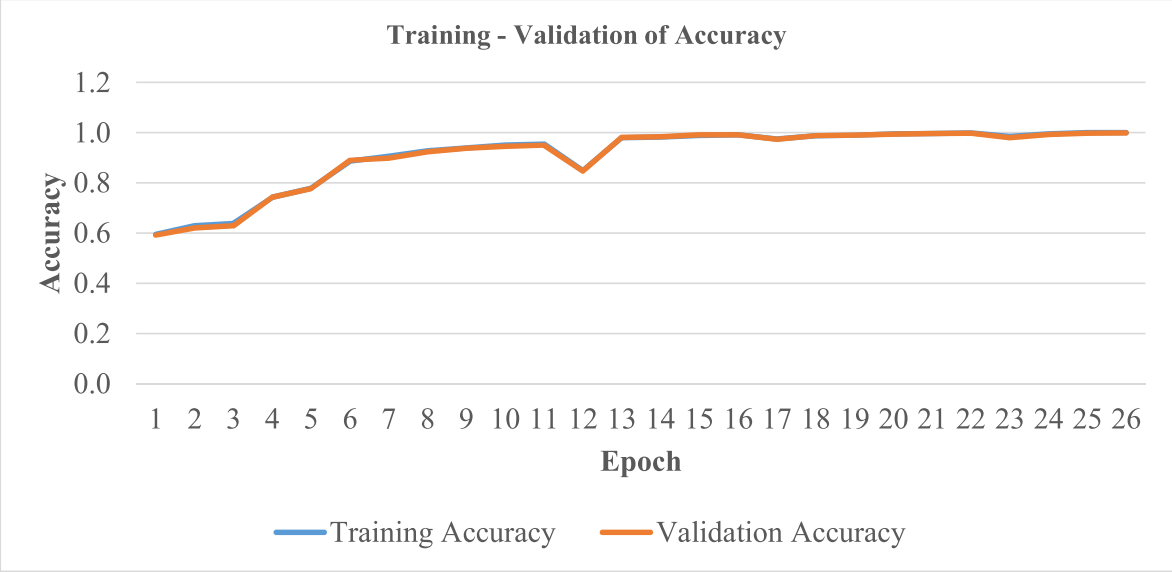


Fig. 11. Training validation accuracy measured during fine-tunning process.

the first training, the score loss is in the higher mode or greater than 0.7, after which the combination of upstream and downstream models can learn and find the global optimum, where the loss can be reduced to close to zero.

For additional guidance, to measure the cross-entropy loss, the accuracy for both was also measured on the training dataset and followed up to validate the model using the validation dataset. Fig. 11 shows the graphical trend of the accuracy score. The horizontal axis represents the number of epochs. The data label detail for Fig. 11 is shown in Table 2. The research fact found that after step 24th epoch, that is steps 25 to 100, the training accuracy archive 1 or 100 %, and when it comes to validation, the accuracy is constant at 0.999 or 99.9%.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$PPV = \frac{TP}{TP + FN} \quad (5)$$

$$TPR = \frac{TP}{TP + FP} \quad (6)$$

$$TNR = \frac{TN}{TN + FP} \quad (7)$$

$$F1\ score = 2 \times \frac{PPV\ x\ TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN} \quad (8)$$

Where:

Acc = Accuracy
PPV = Positive Prediction Value
TPR = True Positive Rate
TNR = True Negative Rate
F1 score = F measure
TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

Table 2. Assessment of training and validation accuracy during the training process.

Epoch	Training Accuracy	Validation Accuracy
1	0,596	0,591
2	0,630	0,620
3	0,638	0,629
4	0,742	0,742
5	0,779	0,777
6	0,887	0,890
7	0,906	0,898
8	0,927	0,924
9	0,939	0,938
10	0,952	0,945
11	0,955	0,950
12	0,849	0,847
13	0,980	0,982
14	0,982	0,984
15	0,989	0,991
16	0,992	0,992
17	0,975	0,974
18	0,987	0,988
19	0,989	0,990
20	0,996	0,995
21	0,997	0,997
22	0,999	0,997
23	0,984	0,980
24	0,996	0,993
25	1	0,998
26	1	0,999
27–100	1	0,999

Table 3. Mapped validation results in a confusion matrix.

		True Label	
		Positive	Negative
Predicted Label	Positive	TP = 763	FP = 0
	Negative	FN = 2	TN = 761

After the merging of the upstream and downstream trains had been completed at 100 epochs, the model of the upstream train and the model of the downstream train were saved. Both pre-training models were then evaluated using the confusion matrix to assess the identification or classification performance. The validation dataset and the test dataset were used to assess the performance of the proposed model. In total, the validation dataset used for evaluation consists of 763 QR code images with positive labels and 763 images with negative or counterfeit labels. As shown in the confusion matrix of Table 3, the result of the validation assessment is as follows, the total number of QR code images with the original positive label that were successfully identified is TP = 763 images and the total number of sample images with the actual positive labels that were detected as negative class is FP = zero. And for the QR code of the counterfeit class that has been successfully detected as a negative image, TN = 761. The number of FN = 2 QR code images with the actual labels is counterfeit has failed to be identified as a negative class. This means that, for the purpose of assessing validation, the detection of a sample from a positive true class could be identified 100% as a positive label.

The accuracy is calculated using the formula in Eq. (4) from the result of the mapped confusion matrix in Table 3. The research fact found the validation accuracy get a score result of 0.9987. The subclass measurement was used to evaluate the positive class accuracy using the precision formula, see Eq. (5) and the results obtained were 0.9974. The assessment also imposed to know the true positive rate, which was mapped in the confusion matrix validation using the sensitivity recall formula, which can be seen in Eq. (6). And for the measurement of the true negative rate, there is the specificity formula, which can be seen in Eq. (7). The result showed that for measuring sensitivity and specificity, the same score of 1 was obtained because the FP score equaled zero. The final assessment for the validation measurement was the F1-score by using the formula shown in Eq. (8), and obtained the mean harmonic score of 0.9987. A brief description of the validation assessment results can be found in Table 4.

Table 4. The validation results.

Validation	Score
Accuracy	0.9987
Precision	0.9974
Sensitivity Recall	1
Specificity	1
F1-Score	0.9987

Table 5. Mapping of test results to a confusion matrix.

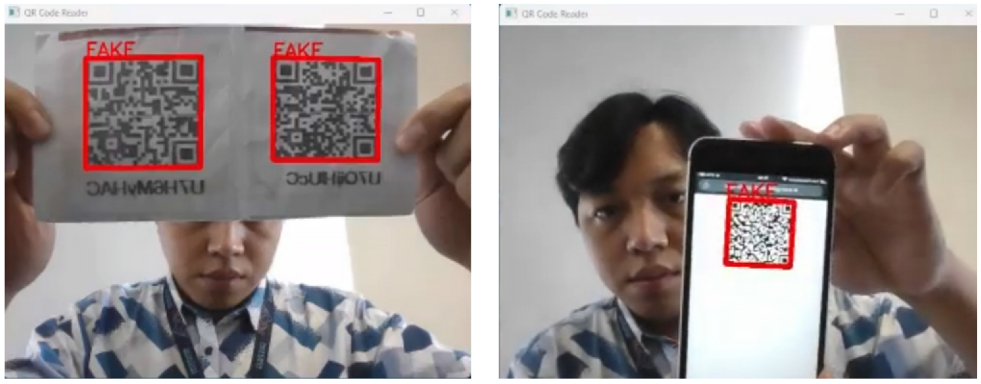
		True Label	
		Positive	Negative
Predicted Label	Positive	TP = 763	FP = 0
	Negative	FN = 7	TN = 756

Table 6. Test score.

Testing	Score
Accuracy	0.995
Precision	0.991
Sensitivity Recall	1.0
Specificity	1.0
F1 Score	0.995

The final evaluation of the classification algorithm is carried out by feeding the model with the sample dataset that has never been tested before. The test dataset was used for its realization. The set of 763 positive class QR codes and 763 negative class QR codes was given to the model individually. Furthermore, the result of the output identification is mapped into a confusion matrix that can be seen in Table 5. All 763 images with positive images were 100% successfully identified as positive class or TP = 763. But it turns out that the entire QR code with the fake tag had only 756 images successfully detected as negative class, leaving 7 images that failed to be detected as negative class or FN = 7.

The plot result of identification proposed model, which is shown in Table 5 from test dataset, then value of TP, FP, FN, and TN was used to measure test result, which is briefly shown in Table 6. As the FP scores are zero in the validated score, the sensitivities and specificities are still similar to the validated score, i.e. 1. In the confusion matrix in Table 5, the increase in the FNs on the test scores had an effect on the scores for accuracy, precision, and F1-score which decreased slightly.



(a) (b)

Fig. 12. The results of the neural embedding implementation using the web camera on laptops. (a) Neural embedding can identify QR codes printed on illegal printers as opposed to legal paper, which can be identified as fake of counterfeit QR codes. (b) The proposed network was also able to identify the QR code displayed using LCD media will be identified as a counterfeit QR code.

The final step was the implementation of the model in a real-world environment, and the results were very positive. The laptop's in-built web camera has been in use. The experiment also involved using the OpenCV Python libraries to access the webcam. Each frame of the video is extracted into images and then fed into the Python library PyZbar to obtain the image area of the QR code. The section of the QR code which has been cropped by the PyZbar, is then fed into the proposed model to determine whether the input image is fake or genuine. The experimental first tried to identify the QR code image printed on the illegal paper by capturing it with the web camera, for a real test using the web camera sensor. As a result, as shown in Fig. 12.b, the proposed model was successful in identifying the QR code as a fake or counterfeit image. In the second real experiment, the input pictures used as QR code pictures were shown from LCD mobile phone media as input, the result can be found in Fig. 12.b where the proposed model can also identify the QR code picture shown from LCD media as a fake image.

The VGG19 architecture's depth and quantity of parameters cause its quite high computational complexity. In real-time applications especially, this might result in more processing time. For example, the VGG19 model calls for large processing resources which may not be practical for IoT systems or mobile devices. Optimization techniques include model quantization, pruning, or lighter architectures like MobileNet or SqueezeNet might be investigated to solve these worries and improve deployment efficiency. However, this is the limitation of our study. Only use computer as a tool to run the program even though the program running well and fast, but when try to deploy in mobile application which has computation limit, the program getting slower an taking few second to detect counterfeit QR code. Real-time counterfeit detection depends critically on the viability of implementing the suggested paradigm in edge devices as IoT systems or cellphones. Effective performance may be obtained even under resource-limited conditions by tweaking the model for reduced computing load. This method will help to allow distributed counterfeit detection, therefore lowering the need for continuous server connectivity and improving the general program responsiveness.

Conclusion

In this study, a novel framework has been proposed that can be used to determine whether the image on a QR code is original or counterfeit. The proposed model consists of an additional convolutional layer to fuse single-layer images into the VGG19 architecture, which requires 3 layers of image input. The modifications have also been targeted at the VGG19 architecture, where the standard downstream layer has not been used. And in exchange, through the adoption of the Siamese triplet network, the VGG19 trained act as a neural embedding extraction feature with the intention of minimizing the intra-class distance between the anchors with original or positive images and increasing the distance interclass between the anchor images with counterfeit or fake images. Once the VGG19 had been trained using the triplet method, the weighting architecture was then fine-tuned by the addition of a proposed downstream architecture. The improvement of the pre-processing on the dataset was also carried out by using Yolo v7 and PyZbar to improve the cropping area and to make the rectification result focus on the image area of the QR code. As a result, the result of

satisfaction is obtained from the validation with an accuracy score of 0.9987 and for the test the score is 0.995. Future research directions may involve the investigation of the implementation of advanced deep learning techniques, the development of additional feature extraction methods, and the expansion of the dataset to improve the model's robustness. Moreover, cooperation with business partners can help to use this technology in practical settings.

Acknowledgment

All resources including financial, computational and dataset to support this study were supported by cooperation between Bina Nusantara University and PT Advotics Teknologi Global.

Authors' declaration

- Conflicts of Interest: None.
- We hereby confirm that all the figures and tables in the manuscript are ours. Furthermore, any figures and images, that are not ours, have been included with the necessary permission for republication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The study was approved by the local ethical committee at Bina Nusantara University, Jakarta, Indonesia, 11480.

Authors' contribution statement

G. P. K. was contributor to the conceptualization and methodology of the study. Y. contributed to methodology, validation, writing paper; R.F. contributed to formal analysis, investigation, and writing original draft preparation; and at Advotics Teknologi Global company, H. C. contributed to resources and supervision; J.W.T. contributed to funding acquisition; P. K. contributed to writing review and editing; M. D. contributed to project management and funding acquisition.

References

1. Scanzio S, Rosani M, Scamuzzi M, Cena G. Codes: From a survey of the state of the art to executable eQR codes for the internet of things. IEEE Internet Things J. 2024;11:23699–23710. https://doi.org/10.1109/JIOT.2024.3385542.
2. Su H, Niu J, Liu X, Atiquzzaman M. SafeCoder: A machine-learning-based encoding system to embed safety identification information into QR codes. J Netw Comput Appl. 2024;227:103874. https://doi.org/10.1016/j.jnca.2024.103874.
3. Li D, Ran X, Jin X. Anti-counterfeit framework of electronic certificate based on QR code and seal watermark. Multimed Tools Appl. Epub ahead of print. 2024;83:80523–80542. https://doi.org/10.1007/s11042-024-18803-x.
4. Gu W, Yin Y, Sun K. An efficient distorted QR code correction method based on improved ResNet architecture. Signal Image Video Process. Epub ahead of print. 2024;18:4223–4231. https://doi.org/10.1007/s11760-024-03066-8.
5. Tran D, De Steur H, Gellynck X, Papadakis A, Schouteten JJ. Consumers' valuation of blockchain-based food traceability: role of consumer ethnocentrism and communication via QR codes. Br Food J. 2024;126:72–93. https://doi.org/10.1108/BFJ-09-2023-0812.
6. Eren BA. QR code m-payment from a customer experience perspective. J Financ Serv Mark. 2024;29:106–121. https://doi.org/10.1057/s41264-022-00186-5.
7. Sahay M, Vanjale S, Mane M. Software as service attack detection and prevention for deceitful QR code. Int. J Intell Syst Appl Eng. 2024;12:454–462. https://doi.org/10.52783/cienceng.v11i1.259.
8. Zheng J, Zhao R, Lin Z, Zhu R, Zhang Z, Fu Y, *et al.* EHFP-GAN: Edge-enhanced hierarchical feature pyramid network for damaged QR code reconstruction. Mathematics. 2023;11:1–19. https://doi.org/10.3390/math11204349.
9. Almousa H, Almarzoqi A, Alassaf A, Alrasheed G, Alsuhibany S. QR Shield: A dual machine learning approach towards securing QR codes. Int J Comput Digit Syst. 2024;15:887–898. https://doi.org/10.12785/ijcds/160164.
10. Masih EA. Feasibility of using QR code for registration & evaluation of training and its ability to increase response rate – The learners' perception. Nurse Educ Today. 2022;111:105305. https://doi.org/10.1016/j.nedt.2022.105305.
11. Picard J, Landry P, Bolay M. Counterfeit detection with QR codes. In: DocEng 2021 - Proceedings of the ACM Symposium on Document Engineering. 2021:1–4. https://doi.org/10.1145/3469096.3474924.
12. Huang Y, Cao P, Li J. Research on multiplexed colour QR code with direct readability. Electron Lett. 2022;58:309–311. https://doi.org/10.1049/ell2.12433.
13. Xie S, Tan HZ. An anti-counterfeiting architecture for traceability system based on modified two-level quick response codes. Electronics (Switzerland). 2021;10:320–331. https://doi.org/10.3390/electronics10030320.
14. Wasnik K, Sondawle I, Wani R, Pulgam N. Detection of counterfeit products using blockchain. ITM Web Conf. 2022;44:1–9. https://doi.org/10.1051/itmconf/20224403015.
15. Shastri SC, Shetty AR, Professor A. Fake product detection using blockchain technology. IJARCCE. 2022;13:2815–2821. https://doi.org/10.55041/IJSREM31871.
16. Bhuiyan MR, Kashem MA, Akter F, Parvin S. Reducing product counterfeiting using blockchain technology in E-Commerce business. Lect Notes Electr Eng. 2023:119–132. https://doi.org/10.1007/978-981-19-8032-9_9.
17. Barrera JF, Mira A, Torroba R. Optical encryption and QR codes: Secure and noise-free information retrieval. Opt Express. 2013;21:5373–5384. https://doi.org/10.1364/OE.21.005373.
18. Li H-S, Chen J, Huang H. QR code arbitrary style transfer algorithm based on style matching layer. Multimed Tools Appl. 2023;83:38505–38522. https://doi.org/10.1007/s11042-023-17231-7.
19. Zhang Z, Yang X, Luo N, Chen F, Yu H, Sun C. A novel method for Pu-erh tea face traceability identification based

on improved MobileNetV3 and triplet loss. Sci Rep. 2023; 13:6986. https://doi.org/10.1038/s41598-023-34190-z.
20. Aljanabi M. Safeguarding connected health: Leveraging trustworthy AI techniques to harden intrusion detection systems against data poisoning threats in IoMT environments. Babylonian Journal of Internet of Things. 2023;2023:31–37. https://doi.org/10.58496/BJIoT/2023/005.
21. Manoharan J, Sivagnanam Y. Enhanced hand gesture recognition using optimized preprocessing and VGG16-based deep learning model. In: 2024 10th international conference on communication and signal processing (ICCSP). IEEE. 2024:1101–1105. https://doi.org/10.1109/ICCSP60870.2024.10543590.
22. Sholihin M, Md Fudzee MF, Ismail MN. AlexNet-based feature extraction for cassava classification: A machine learning approach. Baghdad Sci J. 2023;20:2624–2637. https://doi.org/10.21123/bsj.2023.9120.
23. Yan C, Pang G, Bai X, Zhou J, Gu L. Beyond triplet Loss: Person re-identification with fine-grained difference-aware pairwise loss. IEEE Trans Multimedia. 2022;24:1665–1677. https://doi.org/10.48550/arXiv.2009.10295.
24. Boutros F, Damer N, Kirchbuchner F, Kuijper A. Self-restrained triplet loss for accurate masked face recognition. Pattern Recognit. 2022;124:1–13. https://doi.org/10.1016/j.patcog.2021.108473.
25. Liu L, Dou Q, Chen H, Qin J, Heng P-A. Multi-task deep model with margin ranking loss for lung nodule analysis. IEEE Trans Med Imaging. 2020;39:718–728. https://doi.org/10.1109/TMI.2019.2934577.

اكتشاف قوي لرموز الاستجابة السريعة المزيفة من خلال دمج التضمين العصبي VGG19 والخسارة الثلاثية

جيدي بوترا كوسوم[1]، يوليانت[1]، رينالدي فريديان[2]، هيندي تشاندي[3]، جيفري ويليام تان[3]، فيليب كوان[3]، محمد دافا سيكور[3]

[1] قسم علوم الكمبيوتر، برنامج الدراسات العليا BINUS - ماجستير في علوم الكمبيوتر، جامعة بينا نوسانتارا، جاكرتا، إندونيسيا، 11480.
[2] قسم علوم الحاسوب، كلية علوم الحاسوب، جامعة بينا نوسانتارا، جاكرتا، إندونيسيا، 11480.
[3] أدفوتيك تكنولوجي جلوبال، جاكرتا، إندونيسيا 12950.

الملخص

ان لاكتشاف رموز الاستجابة السريعة المزيفة دورًا حاسمًا في حماية سلامة المنتجات والمستندات. يقدم هذا البحث منهجية جديدة لزيادة موثوقية اكتشاف تزوير رمز الاستجابة السريعة من خلال دمج التضمين العصبي VGG19 والفقدان الثلاثي. يستخدم النهج الشبكات العصبية العميقة ، حيث يتم استخدام تعديل بنية VGG19 لاستخراج ميزات فريدة من رموز الاستجابة السريعة. يتم تضمين هذه الميزات في مساحة متعددة الأبعاد باستخدام طرق التضمين العصبي. يهدف استخدام دالة الفقدان الثلاثية إلى زيادة وضمان القوة التمييزية للتضمينات، وضمان أن تكون تضمينات رمز الاستجابة السريعة الأصلية أقرب إلى بعضها البعض وأبعد عن التضمينات المزيفة.حيث يتم إثبات فعالية ومتانة المنهجية المقترحة من خلال التجارب المكثفة على مجموعة بيانات كبيرة. يحدد النظام المقترح بشكل فعال رموز QR المزيفة بثقة عالية، مع درجات تحقق واختبار تصل إلى 99.87٪ و 99.55٪ على التوالي، حتى عندما تكون الصور مشوهة ومقتطعة ومشوشة. سيتم استكشاف الآثار العملية للبحث، مع تسليط الضوء على التطبيقات المحتملة لمصادقة المنتج، واستراتيجية مكافحة التزوير، والتحقق من المستندات. يؤدي الجمع بين التضمين العصبي VGG19 والفقدان الثلاثي إلى تحسين دقة الكشف، وتعزيز أمان وموثوقية الأنظمة القائمة على رمز QR. يقدم هذا النهج إجابة قوية للمشاكل الصعبة بشكل متزايد للكشف عن التزييف في العصر الرقمي الحديث، وبالتالي يعكس تقدمًا كبيرًا في مجال تكنولوجيا الأمان والمصادقة.كما إنه يحسن موثوقية وسلامة الأنظمة الرقمية بالإضافة إلى حل التحديات التكنولوجية المتعلقة بالمعلومات وتحديد المنتجات المزيفة.

الكلمات المفتاحية: كشف رمز QR، كشف التزييف، التضمين العصبي، نموذج VGG19، الفقدان الثلاثي.