ICIC Express Letters
Volume **17**, Number **8**, August **2023**

ICIC International ©2023 ISSN 1881-803X
pp. **917–925**

QR CODE COUNTERFEIT DETECTION METHOD USING CONVOLUTIONAL NEURAL NETWORKS

Teddy Aristan, Kevin Stephanus Tabaraka and Gede Putra Kusuma

Computer Science Department, BINUS Graduate Program – Master of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ teddy.aristan; kevin.tabaraka }@binus.ac.id; inegara@binus.edu

Received August 2022; accepted November 2022

Abstract. *QR codes are widely known among people, so that it can be used as a place to store product information. However, it turns out QR code can be counterfeited by the perpetrator, so just scanning the QR code is not enough. This paper proposes a method to detect counterfeit QR code based on the appearance of the QR code image. For this study, we evaluated the Convolutional Neural Network (CNN) models used, which are EfficientNetB6, Inception V3, VGG16, VGG19, and ResNet50, to find the best model for detecting QR code counterfeiting. The evaluation was carried out on the dataset which is a combination of genuine and counterfeit QR code images with a total of 522 images. The evaluation results show that the best model for detecting and recognizing counterfeit QR codes is EfficientNetB6, Inception V3, and ResNet50 with an accuracy of 100%. The results of the evaluation prove that there is a method to detect QR codes based only on the appearance of the image and this method is ready to be used because the results are promising.*

Keywords: QR code, Convolutional neural network, Image classification, Counterfeit detection, Transfer learning

1. **Introduction.** A QR (Quick Response) code is a type of matrix barcode or 2D code introduced by the Japanese automotive company Denso Wave in 1994 [1]. A QR code contains the information encoded by text, a URL or other data and consists of black squares arranged in a square grid on a white background. A QR code can be read by devices such as smartphone cameras and decoded very quickly at high speed. A QR code can hold a much larger volume of information: 7,089 characters for numeric only, 4,296 characters for alphanumeric data, and 2,953 binary bytes (8 bits) and 1,817 characters of Japanese Kanji/Kana symbols [2]. A QR code also has error correction capability (by using Reed-Solomon) and the data can be restored even when substantial parts of the code are distorted or damaged [3]. Due to the fast readability and greater storage capacity of QR codes compared to standard UPC (Universal Product Code) barcodes, it became popular and was used for various ways that required the user to retrieve information quickly with the modern devices.

There are many irresponsible people who deliberately counterfeit products and distribute them to the market. These counterfeit products look like the real ones so it is difficult to distinguish. Several attempts have been provided to overcome the problem of product counterfeiting, e.g., by using QR code [4]. By attaching a QR code containing information to the product, it reduces the distribution of counterfeit products in the market [5,6]. People just need to scan the QR code that is attached and information about the product will appear on their smartphone screen, so they can find out the authenticity of the product.

DOI: 10.24507/icicel.17.08.917

However, behind the convenience and the advantages of using QR code, it turns out QR code can be counterfeited by the perpetrator. To enter the market, they attach a counterfeit QR code to the product. Counterfeit QR codes cannot be identified by the human eye and display the same information as the original after scanning it. To create a counterfeit QR code, they perform a print-and-scan attack against the real QR code. A print-and-scan attack is an attack where someone scans an existing image with a scanner and then reprints the image using the different printers. There have been many studies conducted to address this problem, mostly use digital watermarking to detect counterfeit QR codes. An alternative method that can be used is to use a deep learning approach, where this paper proposes an anti-counterfeiting algorithm that can detect and classify the authenticity of products based on QR codes by using Convolutional Neural Network (CNN). We evaluated the five pre-trained CNN models used for this research, which are EfficientNetB6, Inception V3, VGG16, VGG19, and ResNet50. We also performed hyperparameter tuning to find the most optimal hyperparameter and freeze the top layers. The results of these models will be compared to find the best model for QR codes anti-counterfeiting.

This paper is organized as follows. Section 2 presents the related research works. Section 3 discusses the proposed method for this paper. Section 4 presents the experiment from collecting images in a dataset to results from experiments which are displayed in table form and have an explanation. Section 5 shows the conclusions from the results of experiments that have been carried out.

2. **Related Works.** A literature review was conducted to find out previous studies regarding QR code counterfeit detection methods. Xun et al. [7] proposed a dual anti-counterfeiting approach for QR codes that includes information encryption and digital watermarking. The authorization information is encrypted using the RSA-based encryption method. And then, using Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD), they present an anti-print embedding and extraction image watermarking method. To achieve the QR code's dual security, the extracted watermark and decrypted information are compared and verified using cross verification. The results of the attack experiments reveal that digital watermarking has strong robustness and invisibility, and it can implement cross verification successfully through the attack test experiment. The results of the anti-counterfeiting test confirmed the feasibility of the proposed method.

Another method, a new anti-counterfeiting method and a high-reliability digital watermarking method based on DWT and SVD were proposed by Li et al. [8]. The copyright owner's information is embedded as a watermark in the QR code as the carrier image, and they use watermark detection and two-dimensional bar code scanning techniques to extract the watermark information included in the QR code. Experiments have demonstrated that this technique could indeed withstand compressive, rotary, and noise attacks, among other things. This new method and algorithm give the bill of anti-counterfeiting technology a strong guarantee.

In 2017, Liu et al. [9] presented an authentication solution to achieve anti-counterfeiting for messages ciphered using the QR code standard, which includes a secure, visually enhanced QR (SQVR) code, as well as a sensitive authentication method that included digital signature and watermarking method. Safety measures were developed to detect and alert unauthorized modifications. Experiments and analyses demonstrate the effectiveness of the suggested SVQR and verification method. The suggested technique, however, does not support print-and-scan, which seems to be a restriction of using LSB-based watermarking. This problem can be solved with a new watermarking method that supports embedding and extracting watermarks from printed images.

Meanwhile, Nguyen et al. [10] proposed a novel technique for watermarking. They implement a random micro texture (Clipped Gaussian Noise) produced from a 2D random Gaussian signal image. It is used to replace black and/or white QR code blocks. The experiment testing showed that the QR codes cannot easily be counterfeited, due to changes in the behavior of micro textures.

From another research, Wang et al. [11] used infrared watermarking to embed information from an infrared QR code into an explicit QR code. Data hiding with error diffusion is utilized to produce the explicit graphical QR code, with only K can be rendered in infrared. The explicit graphical QR code information can be interpreted using a general QR code reader. The proposed method can be used with existing printing workflows without the use of additional inks or special equipment, and it can also be used in fields related to printing, such as security documents and banknotes.

So far, the papers we have found have all discussed watermarking as a solution to the problem of recognizing counterfeit QR codes. The method without watermark tools is still rarely developed by people. For this study, our main goal is to detect QR code images directly without any tools, so we try to detect QR code with a deep learning approach, such as CNN. CNN is widely used to detect and recognize images (or objects within them) [12] or text [13], predicting data [14], and creating decisions for the application [15]. There are still few papers that discuss QR code anti-counterfeiting with CNN, so we tried to develop and train these several CNN models to serve as a new method for QR code anti-counterfeiting. CNN method and summary for the model used will be discussed in Section 3.

3. **Proposed Methods.**

For the proposed method in this paper, we will use CNN for QR code anti-counterfeiting. Transfer learning is also proposed in this research, to transfer knowledge from a completed model that has been trained with a large dataset [16,17]. For the proposed models, we will use five pre-trained CNN models, which are EfficientNetB6, Inception V3, VGG16, VGG19, and ResNet50, that have been pre-trained with the ImageNet dataset. Using pre-trained models can improve computational efficiency and architectural design [18].

3.1. **EfficientNetB6.**

EfficientNetB6 is one of EfficientNet group models that was proposed by Tan and Le [19]. This model was developed using uniformly scaling up the network's depth, width, and resolution, in which the results obtained have better performance compared to the previous models in accuracy and efficiency. They use neural architecture search as a baseline to design the architecture of EfficientNet family models which is EfficientNetB6. Figure 1 shows the architecture of EfficientNetB6.

3.2. **Inception V3.**

Inception V3 is a development model of Inception that uses less computing power with 42 neural networks that have similar complexity as VGGNet [20]. Inception V3 is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, fully connected layers, and others. Figure 2 shows the architecture of Inception V3.

3.3. **VGG (VGG16 & VGG19).**

Visual Geometry Group (VGG) is a model built by Simonyan and Zisserman [21]. VGG model is a common deep CNN architecture with multiple layers, where the number of multiple layers defines the type of the VGG model, i.e., VGG16 (16 layers) and VGG19 (19 layers). Figure 3 exhibits the architecture of both VGG16 and VGG19.

3.4. **ResNet50.**

ResNet50 is one of the ResNet family models which has 48 convolution layers, 1 max pooling, and 1 average pooling layer (which total is 50 layers) [22]. ResNet50 uses residual learning, where it learns some residuals or can simply be understood as the

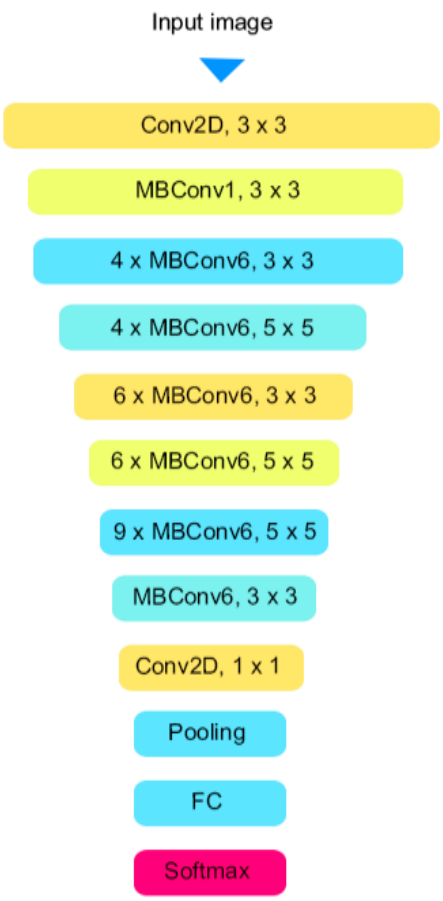

FIGURE 1. Architecture of EfficientNetB6

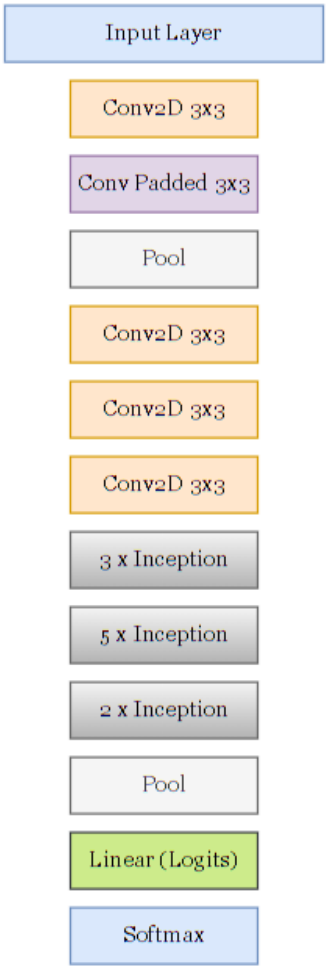

FIGURE 2. Architecture of Inception V3

reduction of features learned from the input of a layer. Figure 4 exhibits the architecture of ResNet50.

3.5. **Model retraining.** As the architectures are shown in figures, many parameters belong to the pre-trained models. However, since the dataset used is not as large as the ImageNet dataset, we will apply a fine-tuning technique in our experiment. We fine tune the five pre-trained models by freezing the top and all the layers except the last two layers in each model. It makes the weight cannot be updated during the training. The purpose of freezing from the initial layer to the last two layers is to prevent all previous learning in the model from being lost. As the replacement for the top layers, we need to add additional layers to predict the output class, i.e., Global Average Pooling and SoftMax layer. Global Average Pooling is a good option to substitute the fully connected network layers where it can be easily interpreted and be directly fed to the SoftMax layer. In the SoftMax layer, the default class is 1000 classes. Henceforth, we changed the class used from default (1000) to 2, for "Counterfeit" and "Genuine" class.

4. **Experiments.** In this section, we will explain about the experiments carried out, from dataset collection, image preprocessing, training, validation, and model testing, and then the results of the experiment.

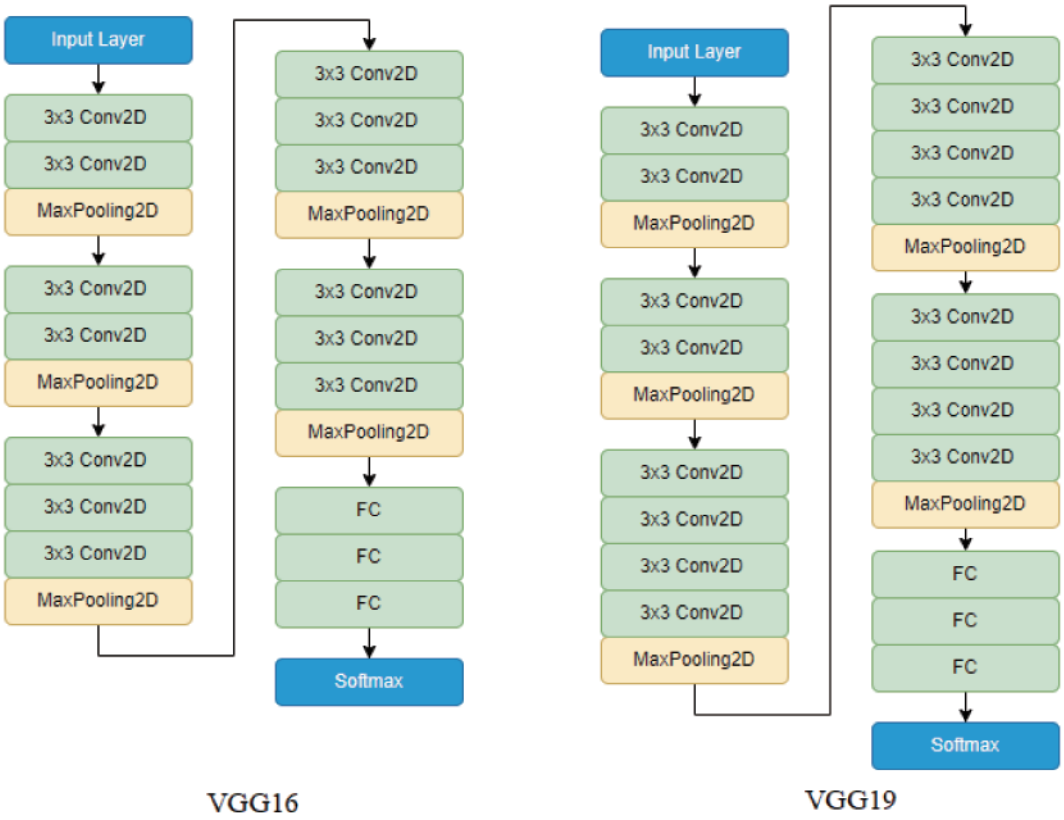

FIGURE 3. Architecture of VGG16 and VGG19

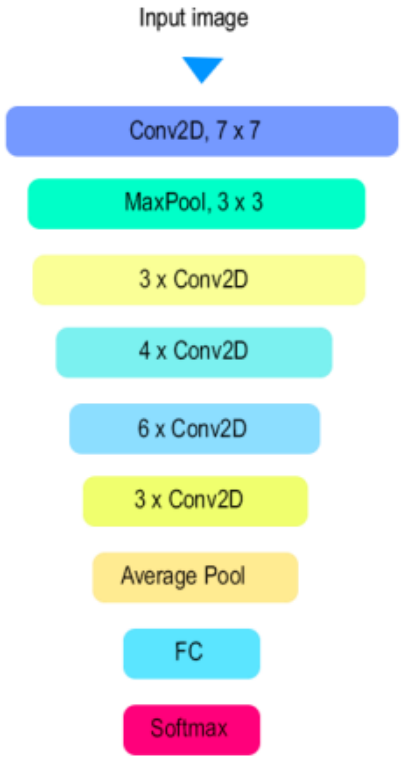

FIGURE 4. Architecture of ResNet50

4.1. **Dataset.** The datasets used for the experiment are ImageNet datasets containing retrified QR code images from PT Advotics Teknologi Global. The collected images contain two classes: "Counterfeit" and "Genuine", each containing 263 and 259 images, then combined into one dataset. Figure 5 shows example of both counterfeit and genuine QR codes for training and testing five pre-trained models.

FIGURE 5. Example of counterfeit and genuine QR codes used for training and testing

4.2. **Experimental design.** Preprocessing the collected data (images) before training our models by using Keras preprocessing function based on each pre-trained model. A dataset total of 522 preprocessing images (from combining "Counterfeit" and "Genuine" dataset) are resized and split into three datasets: the training (312 images), the validation (105 images), and the testing dataset (105 images) by a 60 : 20 : 20 ratio. After we preprocessed images, we checked the models and applied fine tuning to all these models. For fine tuning, it is already explained in Section 3. To save time in training mode, early stopping is also provided, while the training reaches optimal performance. And then, we conduct experiments from training the models to testing the models to measure the performance of each CNN model used. Evaluation measurements will be carried out using 4 metrics: accuracy, precision, recall, and F1 score. Each evaluation metric has a different calculation formula and will be measured in each class that is classified.

4.3. **Experimental results.** In this subsection, it presents the experiment using five pre-trained models on an ImageNet dataset to classify counterfeit and genuine QR codes. All experiments are run in 100 epochs using SGD and Adam optimizer at the learning rate of 0.01 and 0.001. The highest accuracy of each model is determined as the best score. The results of the experiments are arranged in the tables below.

Table 1 represents the result of our five pre-trained models training and validating. The best accuracies for training and validation are Inception V3 and ResNet50 models with 100% accuracy. For best accuracy with fewer data loss, the best CNN model is ResNet50 with data loss of 0.00025 in training and 0.00018 in validation.

TABLE 1. Summary of training and validation results

Model	Training		Validation	
	Accuracy (%)	Loss	Accuracy (%)	Loss
EfficientNetB6	100	0.0209	100	0.0154
Inception V3	**100**	**0.0054**	**100**	**0.0051**
VGG19	50.32	0.6932	50.48	0.6931
VGG16	50.32	0.6932	50.48	0.6931
ResNet50	**100**	**0.00025**	**100**	**0.00018**

Table 2 represents the best hyperparameter of our five pre-trained models. After seeing the results of the comparison on the hyperparameters, we conclude that the best optimizer for the 3 models: EfficientNetB6, Inception V3, and VGG16 is to use SGD, whereas the Adam optimizer is used for the VGG19 and ResNet50 models. All pre-trained models except ResNet used the same learning rate (0.01), whereas ResNet50 uses 0.001. Table 3 to Table 7 represent the confusion matrix of the five pre-trained models testing to know the details.

Table 2. Summary of hyperparameter tuning

Model	Hyperparameter	
	Optimizer	Learning rate
EfficientNetB6	SGD	0.01
Inception V3	SGD	0.01
VGG19	Adam	0.01
VGG16	SGD	0.01
ResNet50	Adam	0.001

Table 3. Testing results by using EfficientNetB6

EfficientNetB6		Predicted	
		Counterfeit	Genuine
Actual	Counterfeit	53	0
	Genuine	0	52

Table 4. Testing results by using Inception V3

Inception V3		Predicted	
		Counterfeit	Genuine
Actual	Counterfeit	53	0
	Genuine	0	52

Table 5. Testing results by using VGG19

VGG19		Predicted	
		Counterfeit	Genuine
Actual	Counterfeit	53	0
	Genuine	52	0

Table 6. Testing results by using VGG16

VGG16		Predicted	
		Counterfeit	Genuine
Actual	Counterfeit	53	0
	Genuine	52	0

The confusion matrix shows how the tested model predicts 105 images from the testing dataset. Of the 105 images, 53 images have the label "Counterfeit" and 52 images have the label "Genuine". From five pre-trained models tested, those that have the best results are EfficientNetB6, Inception V3 and ResNet50, where the predicted label is the same as the actual label (exactly 53 "Counterfeit" and 52 "Genuine").

Table 7. Testing results by using ResNet50

ResNet50		Predicted	
		Counterfeit	Genuine
Actual	Counterfeit	53	0
	Genuine	0	52

Table 8. Summary of testing performances

Model	Testing loss	Testing (%)			
		Accuracy	Precision	Recall	F1 score
EfficientNetB6	**0.013**	**100**	**100**	**100**	**100**
Inception V3	**0.006**	**100**	**100**	**100**	**100**
VGG19	0.694	50.48	25.47	50.47	33.86
VGG16	0.693	50.48	25.47	50.47	33.86
ResNet50	**0.152**	**100**	**100**	**100**	**100**

Table 8 represents the result of our five pre-trained models testing. The worst accuracy for the testing is VGG16 and VGG19 with 50.48% accuracy. Both models cannot be used for authentication because of its very poor performance (small evaluation metrics and high loss). Next, the best accuracies for testing with 100% accuracy are EfficientNetB6, Inception V3 and ResNet50 models. These three models have the same precision, recall, and F1 score, which are 100%. For the data loss, then the best model is Inception V3 with data loss of 0.006 compared to EfficientNetB6 and ResNet50 which have data loss of 0.013 and 0.152.

5. **Conclusions.** In this paper, we have experimented with five pre-trained CNN models on the ImageNet datasets, and then training and testing those models with our custom dataset, as explained in Section 4. Since the available images used are limited and training with deep learning requires a large amount of data, we fine tune five pre-trained models to reduce overfitting issues, so the experiment results can be valid. From the experiment results, CNN models that have the best performance are EfficientNetB6, Inception V3 and ResNet50, both on training and testing. These three models only have a very small difference in data loss, so these three models can be used to identify genuine and counterfeit QR codes. With this developed method, it can have a positive impact, by decreasing product and/or QR codes counterfeiting. For now, there are still few papers that discuss QR code anti-counterfeiting with CNN. In future works, we would like to explore more about CNN for authentication of QR codes or other images (or objects in the image) that are still related to image classification that have an impact on society, i.e., health sector.

Acknowledgment. The authors would like to thank PT Advotics Teknologi Global for datasets which is very helpful for this paper's research.

REFERENCES

[1] J. Rouillard, Contextual QR codes, *Proc. of the 3rd Int. Multi-Conf. Comput. Glob. Inf. Technol. (ICCGI 2008)*, pp.50-55, DOI: 10.1109/ICCGI.2008.25, 2008.

[2] P. Sutheebanjard and W. Premchaiswadi, QR-code generator, *Proc. of 2010 8th Int. Conf. ICT Knowl. Eng. (ICTKE 2010)*, pp.89-92, DOI: 10.1109/ICTKE.2010.5692920, 2010.

[3] S. Tiwari, An introduction to QR code technology, *2016 International Conference on Information Technology (ICIT)*, pp.39-44, DOI: 10.1109/icit.2016.021, 2016.

[4] J. Fei and R. Liu, Drug-laden 3D biodegradable label using QR code for anti-counterfeiting of drugs, *Mater. Sci. Eng. C*, vol.63, pp.657-662, DOI: 10.1016/j.msec.2016.03.004, 2016.

[5] S. Han et al., Lithographically encoded polymer microtaggant using high-capacity and error-correctable QR code for anti-counterfeiting of drugs, *Adv. Mater.*, vol.24, no.44, pp.5924-5929, DOI: 10.1002/adma.201201486, 2012.

[6] M. B. Krishna and A. Dugar, Product authentication using QR codes: A mobile application to combat counterfeiting, *Wirel. Pers. Commun.*, vol.90, no.1, pp.381-398, DOI: 10.1007/s11277-016-3374-x, 2016.

[7] Y. Xun, Z. Li, X. Zhong, S. Li, J. Su and K. Zhang, Dual anti-counterfeiting of QR code based on information encryption and digital watermarking, *Lect. Notes Electr. Eng.*, vol.543, pp.187-196, DOI: 10.1007/978-981-13-3663-8_27, 2019.

[8] D. Li, X. Gao, Y. Sun and L. Cui, Research on anti-counterfeiting technology based on QR code image watermarking algorithm, *Int. J. Multimed. Ubiquitous Eng.*, vol.12, no.5, pp.57-66, DOI: 10.14257/ijmue.2017.12.5.05, 2017.

[9] S. J. Liu, J. Zhang, J. S. Pan and C. J. Weng, SVQR: A novel secure visual quick response code and its anti-counterfeiting solution, *J. Inf. Hiding Multimed. Signal Process.*, vol.8, no.5, pp.1132-1140, 2017.

[10] H. P. Nguyen, F. Retraint, F. Morain-Nicolier and A. Delahaies, A watermarking technique to secure printed matrix barcode – Application for anti-counterfeit packaging, *IEEE Access*, vol.7, pp.131839-131850, DOI: 10.1109/ACCESS.2019.2937465, 2019.

[11] Y. M. Wang, C. T. Sun, P. C. Kuan, C. S. Lu and H. C. Wang, Secured graphic QR code with infrared watermark, *Proc. of the 4th IEEE Int. Conf. Appl. Syst. Innov. (ICASI 2018)*, pp.690-693, DOI: 10.1109/ICASI.2018.8394351, 2018.

[12] L. Shang, Q. Yang, J. Wang, S. Li and W. Lei, Detection of rail surface defects based on CNN image recognition and classification, *Int. Conf. Adv. Commun. Technol. (ICACT)*, pp.45-51, DOI: 10.23919/ICACT.2018.8323642, 2018.

[13] T. Wang, D. J. Wu, A. Coates and A. Y. Ng, End-to-end text recognition with convolutional neural networks, *Proc. of the 21st Int. Conf. Pattern Recognit.*, pp.3304-3308, 2012.

[14] M. Pan et al., Water level prediction model based on GRU and CNN, *IEEE Access*, vol.8, pp.60090-60100, DOI: 10.1109/ACCESS.2020.2982433, 2020.

[15] S. Meister, M. Wermes, J. Stüve and R. M. Groves, Cross-evaluation of a parallel operating SVM – CNN classifier for reliable internal decision-making processes in composite inspection, *J. Manuf. Syst.*, vol.60, pp.620-639, DOI: 10.1016/j.jmsy.2021.07.022, 2021.

[16] L. Torrey and J. Shavlik, Transfer learning, *Handb. Res. Mach. Learn. Appl. Trends Algorithms, Methods, Tech.*, pp.242-264, 2010.

[17] S. Wang, S. Nepal, C. Rudolph, M. Grobler, S. Chen and T. Chen, Backdoor attacks against transfer learning with pre-trained deep learning models, *IEEE Trans. Serv. Comput.*, vol.15, no.3, pp.1526-1539, DOI: 10.1109/TSC.2020.3000900, 2022.

[18] X. Han et al., Pre-trained models: Past, present and future, *AI Open*, pp.225-250, http://arxiv.org/abs/2106.07139, 2021.

[19] M. Tan and Q. V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, *The 36th Int. Conf. Mach. Learn. (ICML 2019)*, pp.10691-10700, 2019.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, Rethinking the inception architecture for computer vision, *Proc. of IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp.2818-2826, DOI: 10.1109/CVPR.2016.308, 2016.

[21] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *The 3rd International Conference on Learning Representations (ICLR 2015)*, https://arxiv.org/abs/1409.1556, 2015.

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