IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 10, 2021, accepted December 9, 2021, date of publication December 13, 2021,
date of current version December 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3135448

Device Light Fingerprints Identification Using MCU-Based Deep Learning Approach

CHUNG-WEN HUNG[1], JUN-RONG WU[1], AND CHING-HUNG LEE[2], (Senior Member, IEEE)
[1]Department of Electrical Engineering, National Yunlin University of Science & Technology, Douliou 640301, Taiwan
[2]Department of Electrical and Computer Engineering. National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan

Corresponding author: Ching-Hung Lee (chl@nycu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under contracts MOST—110-2634-F-009-024, 110-2221-E-A49-121-MY2, 110-2221-E-150-041, 109-2218-E-150-002, and 109-2221-E-224-023, 110-2221-E-224-026, and IRIS and IRIS "Intelligent Recognition Industry Service Research Center" from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

ABSTRACT We introduce device identification using the light fingerprint by a MCU-based deep learning approach. At first, we observe that minor differences exist for individual components of lighting equipment. The corresponding difference produces a unique phenomenon in the frequency spectrum. Therefore, we adopt deep learning approaches for developing a mobile phone light fingerprint identification system and implementing it on a low-cost microcontroller platform. The screen light of the mobile phone is analyzed to obtain the features of unique light fingerprints. We utilize the convolutional neural network, the improved multi-class greedy autoencoder and variational autoencoder with domain adaptation techniques to develop the identification algorithm. Finally, the Bayesian optimization technique is used to optimize the hyper-parameters of models for implementing in the microprocessor. The corresponding comparisons are introduced to demonstrate the performance. The multi-class greedy autoencoder algorithm produces results with an overall accuracy rate and abnormal sample detection rate of 99.67% and 99.85%, respectively. Only a single model needs to be added or deleted for updating new authentication data and this does not affect the identification ability of all models. This results in greater flexibility in real-life applications and potential for expansion to other fields, such as smart buildings and automated robots.

INDEX TERMS Device identification, light fingerprint, machine learning.

I. INTRODUCTION

Recently, the internet of things has introduced new challenges in device identification and authentication [1], [2]. Therefore, light fingerprint applications in device identification are emerging [1]–[3]. Hamidi-Rad *et al.* [3] introduced a 1MHz high-frequency light sensor to collect and transform light source data, and a Raspberry Pi platform was adopted for sampling and performing Fast Fourier Transform (FFT) data pre-processing tasks. Convolution neural network (CNN) and K-nearest neighbors (KNN) algorithms were employed for evaluation. The experiments were conducted using the same brand of LEDs, and the results remained reliable. In the work of Kobayashi [4], the same type of LED was also tested. A photodiode (PD) was used in this study to sense the light and input it through the microphone interface of the cell phone. The algorithm performed FFT preprocessing and then sent the data to a one-dimensional CNN for classification. The experimental recognition accuracy was 97% for 48 LED lighting devices of the same type. A PD was also used to sample the light [5]. The signal was converted to the frequency domain and the Crest Factor (CF) was used as the input feature to identify 331 signals experimentally. After matching, 3000 lamps were tested on site, with an accuracy of 94.4%. These results demonstrated the reliability of using the light frequency or luminous intensity as the identification feature because regardless of the type or number of lamps, differentiation could be accomplished. In addition, positive and negative sample imbalances often occur during the training of machine learning. In [6], owing to the extraction of the current data of a large number of undamaged machining tools and the usage of an autoencoder (AE) to fit the undamaged current data, a reconstruction error arose when abnormal samples were input. Thus, the purpose of detecting damaged tools was achieved, producing experimental results with an accuracy of 95%. In [7]–[10], AE was also used for fault diagnosis, and it was difficult to collect all the samples because of variable fault conditions. Therefore, the training

The associate editor coordinating the review of this manuscript and approving it for publication was Ilsun You.

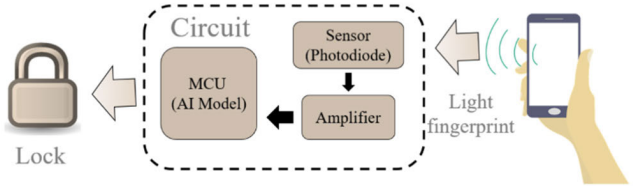

FIGURE 1. System architecture diagram.

method that only fits the positive sample can achieve either normal or abnormal results. However, since AE can only diagnose one condition, the original AE model was improved so that it could simultaneously classify the light source data from multiple cell phones.

In this paper, we utilize the CNN and autoencoder to develop the device light fingerprints identification system in MCU. At first, we observe that there exists minor difference for each individual component of lighting equipment. That is, the corresponding difference causes unique phenomenon in the frequency spectrum for authentication. Therefore, the deep learning approaches are adopted to develop a mobile phone light fingerprint identification system and to implement on a low-cost microcontroller platform. In addition, the Bayesian optimization technique is used to optimize the hyper-parameters of models to implement in the microprocessor. Finally, the corresponding comparisons are shown the performance, the MCGAE algorithm having results with an overall accuracy rate of 99.67% and the abnormal sample detection rate (TPR) of 99.85%.

The remainder of this study is organized as follows. Section II introduces the system structure, includes hardware specification and data acquisition. The proposed device identification algorithm and model optimization are presented in Section III. Section IV introduces the corresponding experimental results. Finally, the conclusions are presented in Section V.

II. SYSTEM STRUCTURE

A. HARDWARE SPECIFICATION

The proposed authentication system architecture using the light fingerprint of a mobile phone is shown in Figure 1, where a PD is used to sense the external light source. A simple amplifier circuit is used to moderately scale the signal detected by the PD, which is then sampled by the artificial intelligence (AI) model installed in the MCU. After sampling, the MCU-based AI model can directly perform the subsequent recognition tasks.

The hardware specifications of the main components for this light fingerprint recognition device were a PD- S6967, amplifier- OPA1612A, and MCU- Renesas RX65N. Thus, the components of the device were simple and inexpensive. The PD (type: S6967) was selected as the sensor for the light source with spectral response range [320 nm, 1060 nm]. After converting the sensed light source into a voltage signal, the voltage range was adjusted by the amplifier circuit. The low noise amplification property of OPA1612A ensured that the small light source features were preserved after amplification. The choice of the MCU was more flexible. Here, the Renesas RX65N with 2 MB of ROM and 640 KB of RAM was adopted to meet the current experimental needs. We successfully realized the technique of converting the artificial intelligence models into C code. The Renesas RX65N is a lower cost MCU that can be selected for actual implementation.

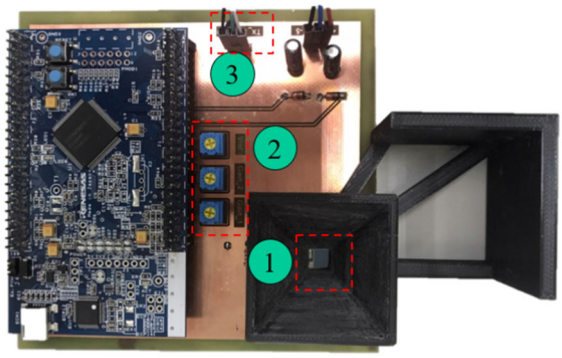

FIGURE 2. Hardware diagram of the proposed approach.

Figure 2 shows the hardware diagram of the proposed approach, where ① denotes the photodiode and a simple shield is used to fix the distance between the phone and the sensor, ② denotes the knob for adjusting the DC level and magnification, and ③ is the UART interface for data collection.

In this study, we first verified the unique features of the light fingerprint, reliability of light fingerprint recognition, and flexibility of the plural light fingerprint recognition system. According to our previous experiences in converting and constructing models in MCUs, the AI models were finally implemented by low-cost MCUs. The processes of the MCU implementation included sampling, FFT transformation, data preprocessing, AI prediction, and control. First, we confirmed the sample acquisition in which the light source was close and sufficiently bright and the MCU triggered the sampling after the PD responded to the voltage change. After sampling, the samples were converted to frequency domain using FFT. Subsequently, the FFT image was re-sized for the AI-model. Finally, the AI Model was trained for prediction.

Herein, the autoencoder (AE) and CNN were utilized to establish the proposed MCU-based AI model. In the AE model, we compared the reconstruction error of each sample model to achieve the individual threshold in the database. If the threshold is reached, we can identify whether the light fingerprint is the one with a known identity. The threshold was determined during the training process and defined in the MCU for identifying the light fingerprint. The CNN model employs a much simpler identification method. The confidence score calculated by the model was compared with the default minimum score. If the score exceeded the default minimum score, it was judged to be a cell phone in the database. Hence, the light fingerprint of mobile phones outside the database can be excluded.

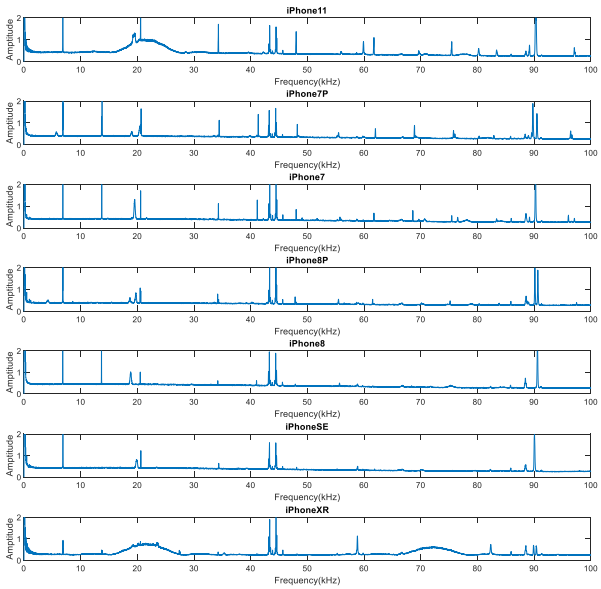

FIGURE 3. Sampling spectrogram of different iPhones.

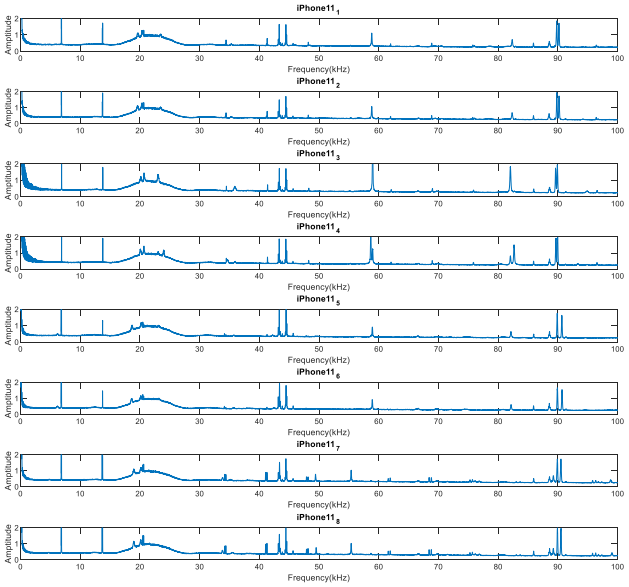

FIGURE 4. Frequency features of different iPhone X1.

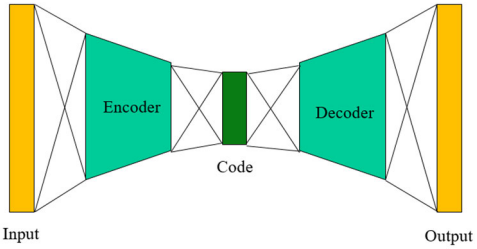

FIGURE 5. Schematic representation of the AE.

B. DATA ACQUISITION

To implement the AI algorithm in MCU, the memory size and computation time of AI model should be considered. Therefore, in the early stage of data collection, a higher spectral resolution and wider spectrum range were adjusted with the oscilloscope to confirm the feature range and select the appropriate spectrum range. In this study, the sampling rate and number of sampling points were 200K sample/second and 16K points, respectively. According to the sampling theorem, the spectral features from 0 Hz to 100 KHz can be obtained with a resolution of 12.2 Hz between frequencies. The actual sampling results (frequency spectrum) are shown in Figure 3 for different models of the iPhone. Figure 3 shows that, the distribution of the spectral features, 500 Hz, 8 KHz, 45 KHz, and 90 KHz, are common and obvious for all models of mobile phones. These features are generated from the sampling circuit and the same component modules used inside the phone. We observed that these features appeared steadily and differences existed between non-iPhone series phones and iPhone series. These have been described in this work.

To illustrate the differences observed in the same model of mobile phones, we collected several iPhone X1 to obtain the corresponding spectral features, as shown in Figure 4. Figure 4 demonstrates the useful features in each area of the sampling spectrum. Each spectrum in Figure 4 is averaged from 100 light fingerprint data of the same phone. Compared with the spectrum of other iPhone series phones, the spectral features of the same model are similar. However, slight differences still occur at 34 KHz, 48 KHz, 60 KHz, 82 KHz, and 90KHz. If the variation phenomenon of light is steady, they can be considered as effective features for distinguishing various cell phones for authentication. Therefore, if these differences can be effectively fitted and reconstructed using AE, the reconstruction error will vary after inputting the same model of different cell phones into the model of each cell phone. The corresponding threshold can then be identified to distinguish between individual cell phones.

III. DEVICE IDENTIFICATION AND MODEL OPTIMIZATION

A. DEVICE IDENTIFICATION USING DEEP LEARNING

Autoencoders (AE) are usually utilized to learn efficient data coding in an unsupervised learning manner [7]–[11]. Its application includes both feature extraction and sample variation diagnosis. It can be broadly divided into two steps, namely data compression and data decompression, as shown in Figure 5. The training aims at making the outputs equivalent to the inputs as much as possible and continuously reduce the reconstruction loss between inputs and outputs. In addition, AE retains data correlation even some information is lost during compressing and uncompressing. Thus, if samples with larger variation are input, the model is unable to effectively recover these samples. Therefore, the reconstruction error increases and this property forms the diagnosis ability described in this study.

If samples not belonging to one of the classes in the original training set are input when ANN, CNN, and other machine learning models are performing the task of classification,

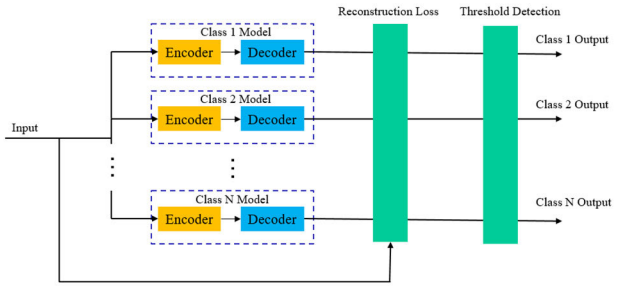

FIGURE 6. Architecture of the MCGAE.

the output will be incorrect, i.e., misclassified as one of the categories. In contrast, the training method of AE, which only fits one type of sample, effectively avoids the aforementioned situation. In general, since the AE detection algorithm cannot support multi-class tasks, therefore, this study improves and extends for a multi-class problem. As shown in Figure 6, a set of AE models are trained independently for each type of sample, and a set of thresholds for reconstruction errors are defined during the training process according to the training conditions. In the actual application, the samples are input to each AE model for individual reconstruction error calculation, and compared to the threshold defined in the training process. If one of the models exhibits a reconstruction error lower than the default threshold, it is identified as this category. Therefore, the estimation of the threshold value is also an important part of this method. In addition, we use a greedy algorithm, called multi-class greedy autoencoder (MCGAE), to search for the optimal threshold,. The greedy algorithm treats all reconstruction errors as thresholds and calculates the accuracy. Therefore, we can obtain the accuracy corresponding to each reconstruction error or determine the proper threshold automatically.

B. MODEL OPTIMIZATION

Here, we describe the optimization of the AI model. We introduce the data collection, list of cell phone models involved in the experiment and their corresponding numbers are listed in Table 1. Nineteen mobile phones were used in our experiments, including eight iPhone X1 models and some iPhone and Android phones. In addition to identifying the differences in features between different phone models, we also explored the ability to distinguish between the same phone models. However, the hyper-parameter affects the accuracy of AI model, several approaches are proposed to optimize model, grid search, random search, and surrogate-based optimization [6], [11], [12]. To obtain the highest accuracy with the smallest model parameters, the Bayesian optimization method was used to search for the best parameters when comparing the models studied in this work. Bayesian optimization (BO) is an approach that used Gaussian process to build a probabilistic model corresponds to the hyper-parameter. By iteratively evaluating based on current model and updating, BO has the ability to build the distribution and objective values when much more information is given [13], [14].

TABLE 1. List of cell phone models.

Phone type (iOS)	number	Phone type (Android)	number
iPhone X1	8	ASUS T100F	1
iPhone 7	1	ASUS T100J	1
iPhone 7Plus	1	Infocus M812	1
iPhone 8	1	Samsung Galaxy A7	2
iPhone 8Plus	1		
iPhone SE	1		
iPohne XR	1		

Herein, the BO is briefly introduced as follows. At first, denote f to be a black-box function without closed-form expression and it is expensive to evaluate. The goal of optimization is to solve the following problem

$$x^* = \arg\min_{x\in\chi} f(x), \quad \chi \in \Re^d \tag{1}$$

where x denotes the hyper-parameters. We here use the BO to find the global optimum by constructing Gaussian process model for $f(x)$ and then exploits to make decisions to next evaluate the function. The pseudo-code of BO procedure is introduced as follows and the optimized algorithm is implemented with "Hyperopt" tool [15].

Algorithm of Bayesian Optimization

Input: initial dada D_0, iterative number
1: **for** $t = 1, 2, \ldots$ **do**
2: Find $x_{t+1} \in \Re^d$ by optimizing the acquisition function $x_{t+1} = \arg\min_{x\in\chi} f(x)$
3: Evaluate $y_{t+1} = f(x_{t+1})$ and augment data $D_{t+1} = D_t \cup (x_{t+1}, y_{t+1})$
4: **end for**
Output: $x_{\max}, y_{\max}$

The corresponding searching parameters for the CNN structure optimization are shown in Table 2. In this optimization, the number of searching parameters was 200 cycles and the optimization of the objective function aimed at estimating the highest accuracy rate of the testing set. The other training parameters were: the number of convolution layers was 2, activation function of convolution layers was set to ReLU, number of training sessions was 1000, learning rate was 0.001, number of batch optimizations was 100, and output activation function was ReLU.

In the AE method, the MCGAE framework shown in Figure 6 was applied. We trained the model variational autoencoder (VAE) using the same training dataset [16], [17]. To reduce the overfitting in the training set, the "Domain adaptation (DA)" concept was utilized to enable the model to focus on the real features of the sample without overfitting the small noise in the training set [18], and the difference between the accuracy of the model after training and the accuracy of

TABLE 2. Hyperparameter searching range for CNN.

Item	Mode	Parameter
Convolution kernel	Choice	1, 2, 4, 8, 16
Kernel size (1*N)	Uniform	2-128
Pooling size (1*N)	Uniform	2-32
Number of fully connected layer	Choice	2, 3, 4, 5
Node of fully connected layer	Uniform	2-64
Activation function (fully connected layer)	Choice	ReLU, Sigmoid, Tanh
Loss function	Choice	MSE, MAE

the actual online test was reduced. Therefore, in this study, we tested the effect of CNN models along with MCGAE, VAE, DA-AE, and DA-VAE for comparison.

Table 3 shows the hyperparameter optimization ranges for MCGAE, VAE, DA-AE, and DA-VAE, respectively. The Bayesian optimization technique was also adopted for hyperparameter tuning. In the hyperparameter search range of these experiments, the model was designed by setting ''Number of hidden layers'' and ''Maximum of nodes in the hidden layer'' in a 100% incremental or 100% decremental manner every other layer. This method effectively controls the model size and adds a dynamic dropout layer with Hyperparameter tuning 200 times. The other training parameters were assigned as follows: Learning rate 0.001, Epoch 1000 times, Dropout rate 0.2, and Optimizer Adam. Note athet the DA methods, DA-AE and DA-VAE, whose structural design is presented in Figure 7. The testing data was viewed as the target domain. The new debugging parameters werethe part of the domain classifier, which contained the maximum number of nodes, layers, and activation function. In the models of MCGAE, VAE, DA-AE, and DA-VAE, the greedy algorithm was used to automatically test all threshold values to find the one with the highest accuracy rate.

C. IMPLEMENTATION OF AI MODEL INTO MCU

Most of the recently AI model were implemented in PC or cloud computational system. This results high-cost and communication loss problems, therefore, we here introduce the implementation of AI model into microcontroller unit (MCU) for device identification. The trained model should be translated to C code. The most common approaches relate to model conversions for AI model frameworks such as Caffe 2, TensorFlow Lite for microcontroller, and Arm NN for deploying trained models and inference engines on MCUs [19]. There are software tools that take pre-trained AI models for MCUs by converting them into C-code. This paper utilizes Renesas RX65N MCU for implementation, therefore, the corresponding e-AI translator is adopted [20], that is, e-AI converses the trained AI model into C-language code for RX65N MCU. Moreover, the tool also performs the calculations of the memory size and amount of calculation required by AI model are also estimated, and the multiply and accumulation number calculation when the AI model operates.

TABLE 3. Hyperparameter searching range.

Item	Mode	Parameter
MCGAE		
Number of hidden layer	Choice	3, 5, 7, 9, 11, 13, 15
Maximum of nodes in the hidden layer	Uniform	2~8
Activation function (hidden layer)	Choice	ReLU, Tanh, Sigmoid
Loss function	Choice	MAE, MSE
VAE		
Number of hidden layer	Choice	3, 5, 7, 9, 11, 13, 15
Maximum of nodes in the hidden layer	Uniform	2~8
Nodes number of noise code	Uniform	2~8
Activation function (hidden layer)	Choice	ReLU, Tanh, Sigmoid
Loss function	Choice	MAE, MSE
DA-AE		
Number of hidden layer	Choice	3, 5, 7, 9, 11, 13, 15
Maximum of nodes in the hidden layer	Uniform	2~8
Activation function (hidden layer)	Choice	ReLU, Tanh, Sigmoid
Hidden layers number of DA	Choice	3, 5, 7, 9, 11, 13, 15
Maximum number of DA nodes	Uniform	2~8
Activation function (DA)	Choice	ReLU, Tanh, Sigmoid
Loss function	Choice	MAE, MSE
DA-VAE		
Number of hidden layer	Choice	3, 5, 7, 9, 11, 13, 15
Maximum of nodes in the hidden layer	Uniform	2~8
Activation function (hidden layer)	Choice	ReLU, Tanh, Sigmoid
Nodes number of noise code	Uniform	2~8
Maximum number of DA nodes	Uniform	2~8
Activation function (DA)	Choice	ReLU, Tanh, Sigmoid
Loss function	Choice	MAE, MSE

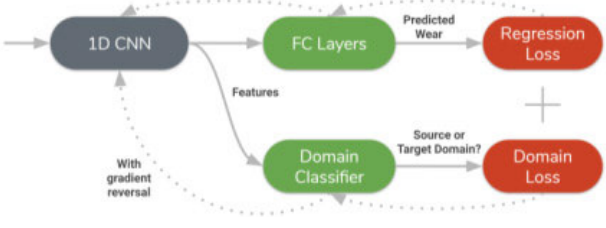

FIGURE 7. Domain adaptation learning architecture diagram [18].

IV. EXPERIMENTAL RESULTS

To facilitate the discussion, ''normal samples'' will be used in this section to denotes the cell phone samples involving training, and ''abnormal samples'' indicates the cell phone samples not participating in training. An ''abnormal sample'' is a cell phone sample that did not participate in the training.

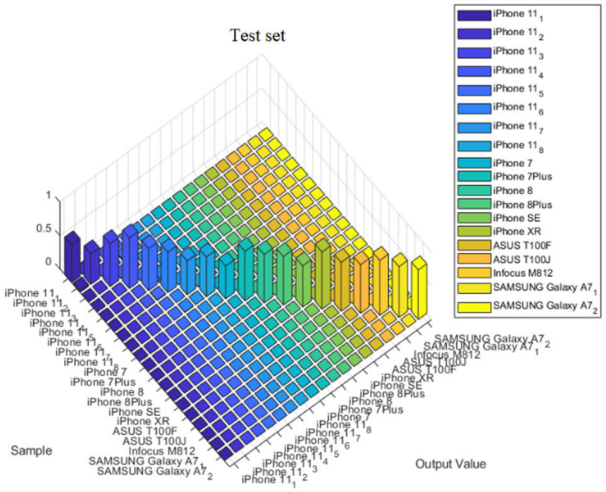

FIGURE 8. Training results of the CNN model.

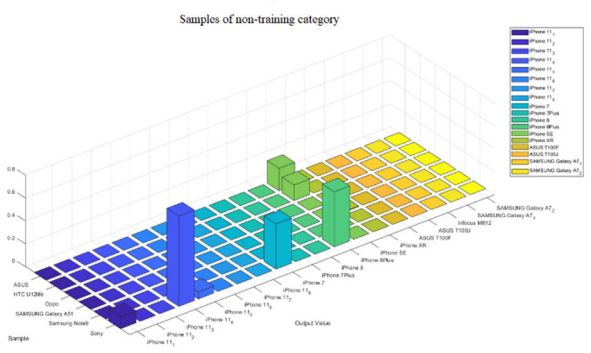

FIGURE 9. Test results for the abnormal sample using the CNN model.

TABLE 4. Summary of results for AE-related methods.

Model	Priority selection	Testing set TPR	Testing set ACC	Size (KB)
MCGAE	Best ACC	99.85%	99.67%	112
MCGAE	Best TPR	99.85%	99.67%	112
VAE	Best ACC	99.41%	98.99%	112
VAE	Best TPR	99.85%	98.92%	112
AE-DA	Best ACC	99.81%	99.06%	144
AE-DA	Best TPR	99.95%	98.84%	144
VAE-DA	Best ACC	99.71%	98.46%	144
VAE-DA	Best TPR	99.92%	98.31%	144

The True Positive Rate (TPR) of abnormal samples is also a very important part of this application. Therefore, in addition to the accuracy rate as the target of training, improving the TPR of abnormal samples is also essential.

In the CNN model, ReLU was chosen as the output activation function and a set of threshold scores was determined for each category according to the training results. The training results are shown in Figure 8, which shows the predicted confidence scores for each category of samples input to the CNN model. The terms ''Sample'' and ''Output Value'' denote the sample of each model and output category of the model, respectively. For each version, 100 testing set samples were tested and the results were averaged and plotted on the graph. The accuracy of the testing set was 100%, and the confidence scores of each category were very clear, without overfitting.

Based on the above CNN results, only the predictive effect of normal samples was tested. However, the performance of abnormal samples was also the focus of this study. Figure 8 shows the results of testing on the trained CNN model with abnormal samples, using the light fingerprint data from six untrained phones as the abnormal samples. The prediction result of the Sony model in Figure 9 shows that the confidence score of the iPhone X1 is higher than 0.6, and the confidence scores of the iPhone 7 and iPhone 8 Plus are also predicted to be very high. In comparison with the results of Figure 8, where the confidence scores of all samples are approximately zero except for the category of correct samples, the performance of CNN in detecting abnormal samples in the non-training set is relatively poor.

The detailed results of each algorithm in the AE-related method are summarized in Table 4. In this experiment, MCGAE performs the best after comparing the algorithms. MCGAE uses the most original AE structure and exhibits better ACC and TPR in both training and testing sets than other deformation methods, indicating that there is no significant difference between the training and testing sets in this application. The VAE and DA methods attempt to reduce overfitting and increase generality by restricting the model and adding an adversarial structure to the model, which may affect the overall accuracy of the model. However, if there is a large discrepancy between the performances of the training and testing sets, it is possible to exert its effect and improve the accuracy of the testing set. As our experiences, we can observe that there is hardly variation in training and testing error using different Autoencoder methods due to domain-adaptation learning. It also demonstrates the effectiveness.

V. CONCLUSION

This study utilizes various Artificial Neural Network algorithms. The comparative algorithms developed for the light fingerprint identification system include CNN, AE, and VAE, as well as the introduction of DA to reduce the distribution distance between the source and target domains to reduce the overfitting of the models. All models were finally installed on the Renesas RX65N MCU. Therefore, during the training process, GS or Bayesian parameter optimization methods were used to optimize the search. The model with higher accuracy and lower memory requirement was selected from the search results for subsequent evaluation.

FFT was used as the pre-processing algorithm for the data to compare the effectiveness of machine learning models such as CNN, MCGAE, VAE, DA-AE, and DA-VAE. In this experiment, the Bayesian method was used to optimize the hyperparameters. Among the five algorithms, CNN is a classification algorithm. In the training and validation sets, a 100% recognition rate can be achieved. However, since it could not be effectively detected on the untrained testing

samples, it did not meet the requirements of this application. The other four algorithms were variations of the AE algorithm. Among them, MCGAE exhibited the best performance with a 99.85% detection rate of abnormal samples and a 99.67% overall accuracy rate. The feasibility of artificial intelligence to identify light fingerprint anomalies was verified through the empirical results in this paper,. The concept of light fingerprint can be applied applications of authentication.

REFERENCES

[1] Y. Liu, J. Wang, J. Li, S. Niu, and H. Song, "Machine learning for the detection and identification of Internet of Things (IoT) devices: A survey," *IEEE Internet Things J.*, early access, Jul. 21, 2021, doi: 10.1109/JIOT.2021.3099028.
[2] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "IoTSense: Behavioral fingerprinting of IoT devices," 2018, *arXiv:1804.03852*.
[3] S. Hamidi-Rad, K. Lyons, and N. Goela, "Infrastructure-less indoor localization using light fingerprints," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 5995–5999.
[4] H. Kobayashi, "CEPHEID: The infrastructure-less indoor localization using lighting fixtures' acoustic frequency fingerprints," in *Proc. 45th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Lisbon, Portugal, Oct. 2019, pp. 6842–6847.
[5] C. Zhang and X. Zhang, "Pulsar: Towards ubiquitous visible light localization," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw. (MOBICOM)*, vol. 17. New York, NY, USA: Academy of Medicine, Oct. 2017, pp. 208–221.
[6] C.-W. Hung, W.-T. Li, W.-L. Mao, and P.-C. Lee, "Design of a chamfering tool diagnosis system using autoencoder learning method," *Energies*, vol. 12, no. 19, p. 3708, Sep. 2019, doi: 10.3390/en12193708.
[7] Z. Xu, W. Mo, Y. Wang, S. Luo, and T. Liu, "Transformer fault diagnosis based on deep brief sparse autoencoder," in *Proc. Chin. Control Conf. (CCC)*, Guangzhou, China, Jul. 2019, pp. 7432–7435, doi: 10.23919/ChiCC.2019.8866347.
[8] K. Yang, Y. Ma, and Y. Han, "Intelligent fault diagnosis of bearing based on stacked denoising autoencoders," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Internet Things (CCIOT)*, Dalian, China, Oct. 2018, pp. 346–350, doi: 10.1109/CCIOT45285.2018.9032655.
[9] Y. Zeng, X. Wu, and J. Chen, "Bearing fault diagnosis with denoising autoencoders in few labeled sample case," in *Proc. 5th IEEE Int. Conf. Big Data Anal. (ICBDA)*, Xiamen, China, May 2020, pp. 349–353, doi: 10.1109/ICBDA49040.2020.9101321.
[10] I.-H. Kao, W.-J. Wang, I.-C. Chiang, and J.-W. Perng, "Implementation of permanent magnet synchronous motor fault diagnosis by a stacked autoencoder," in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, Taichung, Taiwan, May 2018, pp. 1–2, doi: 10.1109/ICCE-China.2018.8448953.
[11] C.-W. Hung, S.-X. Zeng, C.-H. Lee, and W.-T. Li, "End-to-end deep learning by MCU implementation: An intelligent gripper for shape identification," *Sensors*, vol. 21, no. 3, p. 891, Jan. 2021, doi: 10.3390/s21030891.
[12] H.-Y. Chen and C.-H. Lee, "Vibration signals analysis by explainable artificial intelligence (XAI) approach: Application on bearing faults diagnosis," *IEEE Access*, vol. 8, pp. 134246–134256, 2020.
[13] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016, doi: 10.1109/JPROC.2015.2494218.
[14] V. Nguyen, "Bayesian optimization for accelerating hyper-parameter tuning," in *Proc. IEEE 2nd Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Jun. 2019, pp. 302–305.
[15] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 115–123.
[16] J. An and C. Sungzoon, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lect. IE*, vol. 2, no. 1, pp. 1–18, 2015.
[17] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," 2016, *arXiv:1609.08976*.
[18] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.
[19] *TensorFlow Lite for Microcontrollers*. Accessed: Apr. 17, 2021. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers
[20] *e-AI Solution e-AI Translator Tool*. Accessed: Aug. 20, 2020. [Online]. Available: https://www.renesas.com/jp/en/solutions/key-technology/e-ai.html

CHUNG-WEN HUNG received the Ph.D. degree in electrical engineering from the National Taiwan University, in 2006. He is currently a Professor with the National Yunlin University of Science & Technology. His research interests include the IoT, IIoT, power electrelectronics, motor control, and AI application.

JUN-RONG WU received the B.S. degree in electrical engineering from the Southern Taiwan University of Science and Technology and the master's degree in electrical engineering from the National Yunlin University of Science and Technology, in 2021. His research interests include data science and artificial intelligence algorithms in automatic systems.

CHING-HUNG LEE (Senior Member, IEEE) was born in Taiwan, in 1969. He received the B.S. and M.S. degrees from the Department of Control Engineering and the Ph.D. degree from the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1992, 1994, and 2000, respectively. He is currently a Professor with the Department of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Hsinchu. His research interests include artificial intelligence, fuzzy neural systems, neural networks, signal processing, nonlinear control systems, robotics control, and CNC motion control and optimization. He received the Wu Ta-Yu Medal (Young Researcher Award) from the Ministry of Science and Technology, Taiwan, China, in 2008. He also awarded the fellow, Youth, and Excellent Automatic Control Engineering Awards from Chinese Automatic Control Society Taiwan at 2019, 2009, and 2016, respectively.