Liveness Detection Based on Improved Convolutional Neural Network for Face Recognition Security

Article · August 2022

DOI: 10.46338/ijetae0822_06

EXPLORING RESEARCH AND INNOVATIONS

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 12, Issue 08, August 2022)

Manuscript Received: 26 June 2022, Received in Revised form: 20 July 2022, Accepted: 01 August 2022 DOI: 10.46338/ijetae0822_06

Liveness Detection Based on Improved Convolutional Neural Network for Face Recognition Security

Yang Wei[1], Ivy Kim D. Machica[2], Cristina E. Dumdumaya[3], Jan Carlo T. Arroyo[4], AllemarJhone P. Delima[5]

[1,2,3]*College of Information and Computing, University of Southeastern Philippines, Davao City, Davao del Sur, Philippines*
[1]*College of Big Data, Baoshan University, Baoshan, Yunnan, China*
[4,5]*College of Information and Computing Studies, Northern Iloilo State University, Estancia, Iloilo, Philippines*
[4]*College of Computing Education, University of Mindanao, Davao City, Davao del Sur, Philippines*

***Abstract*—Face liveness detection is an important biometric authentication method for face recognition securitythat is used to determine a fake face from an authentic one. In this paper, a liveness detection method based on optimized LeNet-5 is proposed. The LeNet-5 is optimized by increasing the convolution kerneland byintroducing a global average pooling. The simulation results show that the proposed model obtained the highest recognition rate of 99.95% as against the 96.67% and 98.23% accuracy from the Support Vector Machine (SVM) and LeNet-5 models, respectively.The results denote that the proposed model has a high recognition rate in face liveness detection.**

***Keywords*—Convolutional neural network; face recognition security; improved LeNet-5; liveness detection**

I. Introduction

A well-known security process used to protect access to digital computing devices is through biometric authentication [1]. The authentication system determines the individual's identity based on biological characteristics that are unique to the individual [2]. Some of the popular authentication schemes include hand and finger geometry, face recognition, voice identification, vein geometry, iris recognition, retina scan, speaker recognition, and fingerprint scan, among others [3].

Among these, one of the most popular biometric authentication methods used for identity management and secure access control for many weband mobile-related software applications is the face recognition[4]. Face recognition is more convenient to deploy than any other biometric technique. However, despite its advantage as a non-intrusive form of access, the security system might not be able to distinguish between a real person and his or her photograph. An impostor can gain access to the system by presenting a copy of the image to the camera.

Therefore, prior to face recognition authentication, face liveness detection should be implemented to detect whether the captured face is live or fake[5]. To address face spoofing attacks, researchers have proposed different methods for face liveness detection, such as the use of the enhanced local binary pattern (LBP), motion analysis, texture analysis, and quality analysis of captured images, among others. However, recent research has focused on using deep convolutional neural network (CNN) architectures for the face liveness detection[6], where accuracy could still be improved.

This paper studies the security problems in face authentication and focuses on several key technologies in security authentication. Further, a liveness detection method based on optimized LeNet-5 [7] is proposed.Taking advantage of CNN's automatic image feature extraction, the human face image and photo printed face image are classified. The face image features are obvious, which makes it easier to distinguish a living face from a nonliving face.The main work of this paper focuses on the structure, convolution kernel size, number of feature maps, and full connection layer of LeNet-5. The model details are studied and discussed, and a living classification model is proposed.

II. Relative Works

A. Face Recognition

Facial images are the most common biometric characteristic used by humans to make a personal recognition. The most popular approaches to face recognition are based on either: 1) the location and shape of facial attributes such as eyes, eyebrows, nose, lips, and chin, and their spatial relationships, or 2) the overall (global) analysis of face image that represents a face as a weighted combination of a number of canonical faces [8].

The accuracy of face authentication has improved by leaps and bounds, far exceeding the recognition ability of human eyes. Therefore, face authentication is more and more applied to various actual scenes [9], such as file management, human-computer interaction, criminal identification, credit card verification, and access control system, among others. With the development of mobile terminal technology, online payment has become the conventional mode of online shopping payment.

With the popularity of face authentication systems, face information should also be secured. Therefore, when improving the accuracy of the face recognition system, its security should also be strengthened. At present, there are still great security vulnerabilities in the face authentication system. How to improve the security of the face authentication system and improve the security of users' property and privacy is not only an urgent problem to be solved but also a prerequisite for the further popularization of the face authentication system. The security of face authentication systems has attracted more and more attention from experts and research institutions locally and in abroad.

B. Face Recognition Modules

A typically face recognition system can be divided into four stages [4], [10]-[14], such as:

- *Acquire image*

In this process, one or more faces are discovered in an image or a video.

- *Feature extraction*

The facial features are extracted for the next matcher task.

- *Matcher*

Matcher module that matches the features extracted from the above-mentioned module.

- *Decision*

Software decides whether or not any comparisons from the matcher in step 3 are close enough to declare a possible match.

C. Attack on Face Recognition

Several studies have analyzed the likelihood of security breaches and the potential approaches to counter these vulnerabilities. The study of[15] identified the potential points of an adversary attack on a biometric system. These attacks are applicable to any information system, including a face recognition system.

The attacks using fake biometrics and template modification are unique to the biometric system. The characteristics of such attacks [16]-[20] are as follows:

(1) *Fake face:* Illegal attackers use the fake face of genuine users, such as fake images and videos, to invade the system. With the development of the social network, the face photos or videos of legal users are easy to be obtained by criminals to invade the system to achieve some purpose. Therefore, it is necessary to design an effective algorithm to judge whether the requester is a real or a fake face.

(2) *Replay attack:* Such attacks are often used in the data transmission channel between image acquisition and feature extraction, and pre-injected biometrics are used to replace the biometrics sample. An effective countermeasure against this kind of attack is to time mark each request.

(3) *Tamper feature extractor:* It is often used in the feature extraction module to make the feature extractor generate a specific feature vector.

(4) *Synthetic feature vector:* Such attacks are often used in the data transmission channel between feature extraction and similarity matching, and the synthetic feature vector is used to replace the real feature vector.

(5) *Modify similarity matching value:* This kind of attack is often used in similarity matching module, which adopts a fake matcher to output a higher matching value.

(6) *Tamper face template:* Such attacks are often used to attack the face template database, such as modifying or deleting the template in the database, or adding some templates stored in other databases. The most serious consequence is the cross-matching of different databases, that is, stealing the template in the database for other purposes, such as stealing the face template from the bank database to obtain someone's health record, and the like. The related research on this problem is biometric template protection.

(7) *Attack storage channel:* This kind of attack is often used in the transmission channel between database and similarity matching. Another template is used to replace the template input matcher stored in the database.

(8) *Attack decision strategy:* This kind of attack is often used in the authentication decision module, using random numbers to replace the response value of the matcher. Because the output of traditional face authentication is 0 or 1, it can obtain a 50 percent success rate.

D. *Face Liveness Detection*

Liveness detection is a mechanism that is used to detect the input sample feature that is provided by a live human being. It is an ability to distinguish between a real input sample feature provided by the living human being and a fake input feature provided by an artifact. Liveness detection can be applied using software or hardware means. The use of extra hardware to implement liveness detection is to measure various life signs like pulse detection, blood pressure, movement of the face, and eyes for the face recognition [21]. However, one limitation of using extra hardware is the high cost.

III. METHODOLOGY

A. *Convolutional Neural Network (CNN) Model*

A typical CNN consists of the following parts: input layer, convolutional layer, pooling layer, full connection layer, and output layer. The convolution layer is used to extract image features, whereas the pooling layer is used to reduce the number of parameters, and the full connection layer is used to output the results. Usually, CNN contains a series of convolution layers and pooling layers. Thegraphical representation of a typical simple convolution neural network is shown in Fig. 1. The convolution layer, pooling layer, and full connection layer are introduced below.

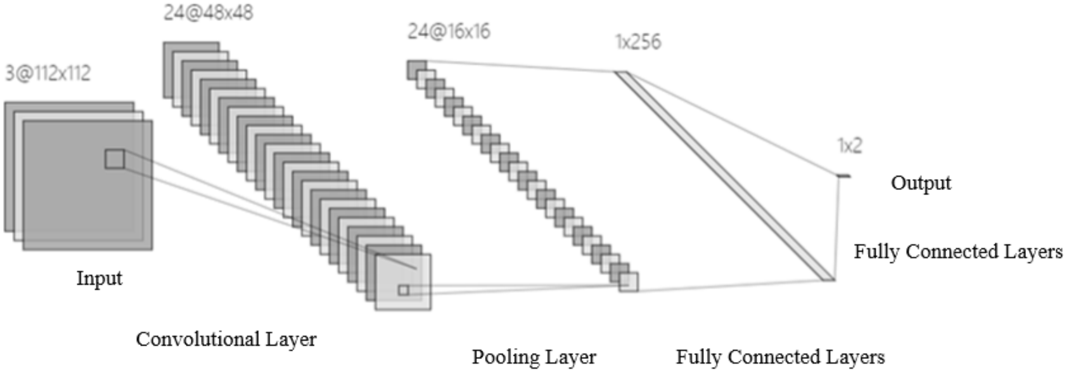


Fig. 1. Simple CNN composition diagram

1. Convolutional Layer

The core of CNN is the convolutional layer, which performs an operation called "convolution". In neural networks, the purpose of the convolution layer is to extract features. The convolution layer convolutes the input features and transmits the results to the next layer. The convolution calculation process is shown in Fig. 2.

The work of the convolution operation is to multiply the convolution kernel (filter) with the input. The size of the convolution kernel is smaller than the input data, and the multiplication type between the input and convolution kernel is the product, that is, element by element multiplication between the input and convolution kernel, and then sum to obtain a single value. The convolution kernel is systematically applied to each small area of the input data with the same size as the convolution kernel, from left to right, from top to bottom, and then the corresponding characteristic graph is obtained. The convolution operationchanges the characteristic graph.

The process is: suppose the shape of the input tensor is: the number of images × Image width × Image height × Image depth, the shape of the feature map after convolution layer is: number of images × Feature width × Height of feature map × Signature channel.

2. Pooling Layer

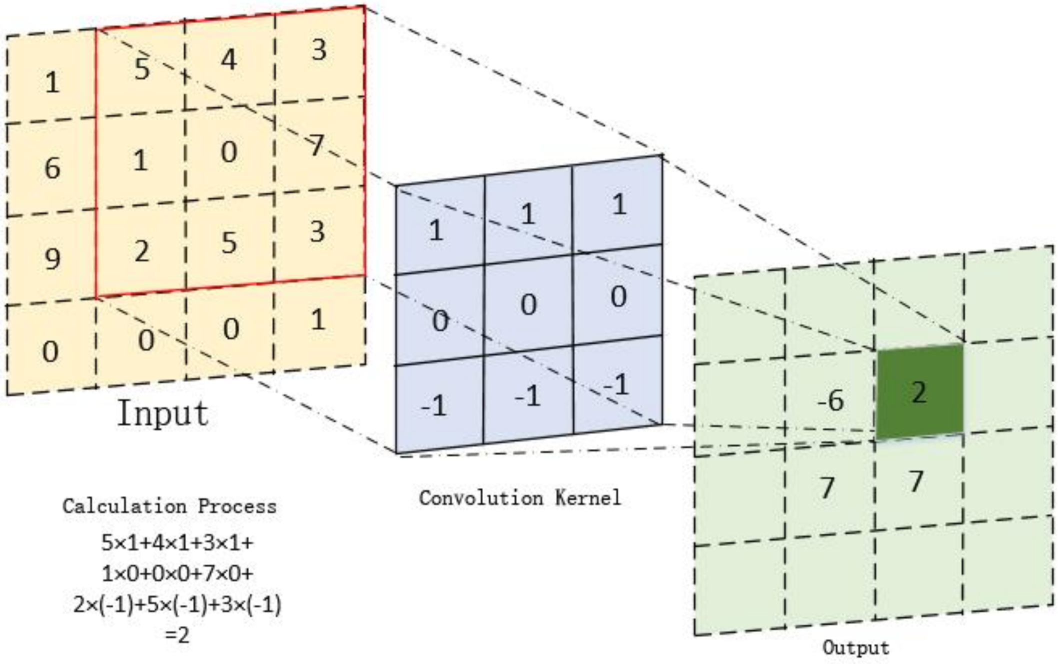
The pooling layer is also an important part of the convolutional neural network. It usually follows the convolution layer to aggregate the information of the features extracted by the convolution layer, so as to reduce the data dimension. The pooling layer is used to reduce the dimension of data by simulating the human visual system. For a large feature map, aggregate and count different regions, such as calculating the maximum or average value of the region, so that the image can be represented by higher-level features. The calculation process of pooling is similar to convolution, sliding on the feature map through a sliding window.

Common pooling operations include maximum pooling and average pooling, which are used to describe the most active performance and average performance of features, respectively. Maximum pooling is instrumental to find the maximum value in each region, while average pooling is used to average all elements in the corresponding region.

3. Fully Connected Layers

In CNN, the full connection layer is usually behind multiple convolution layers and pooling layers, which play the role of classifying the previously learned features. Its structure is similar to the multi-layer perceptron model.

The neurons of each layer in the full connection layer is fully connected with the neurons of the next layer, which can effectively integrate the differentiated feature information learned in the convolution layer and pool layer. Compared with the previous direct input of the whole image to the multi-layer perceptron, through the dimensionality reduction of the data by the convolution layer and pooling layer, the full connection layer can effectively use these purified features. For a classification task, the output value of the last layer of the full connection layer is generally processed by logistic regression, and the final output vector represents the probability that the input feature belongs to a label.



1	5	4	3
6	1	0	7
9	2	5	3
0	0	0	1

1	1	1
0	0	0
-1	-1	-1

	-6	2	
	7	7	

Calculation Process
5×1+4×1+3×1+
1×0+0×0+7×0+
2×(-1)+5×(-1)+3×(-1)
=2

Fig. 2. Process of convolutional calculation

B. LeNet-5

The LeNet-5 is one of the classic CNN models in deep learning. It is a CNN with a relatively simple structure. It was first used to recognize handwritten datasets. The LeNet-5 mainly includes 7 structural layers, including 3 convolution layers (C1, C3, C5), 2 pooling layers (S2, S4), 1 full connection layer (F6), and 1 output layer. In addition, the LeNet-5 also includes an input layer with a picture input size of 32 × 32.

The core size of C1 is 5 and the step size is 1. After the image of the input layer is operated by this layer, six 28×28 feature images will be the output.

The core size of S2 is 2 and the step size is 2. The characteristic diagram output by C1 will output six 14×14 characteristic diagrams after the operation of this layer. Further, the core size of C3 is 5 and the step size is 1. The characteristic diagram output by S2 will output 16 10 ×10 characteristic diagrams after the operation of this layer. Furthermore,the core size of S4 is 2 and the step size is 2. The characteristic diagram output by C3 will output 16 5×5 characteristic diagrams after the operation of this layer. The core size of C5 is 5 and the step size is 1. The characteristic diagram output by S4 will output 120 1×1characteristic diagrams after the operation of this layer.

The F6 is the full connection layer, with a total of 84 neuron nodes, and each neuron in the layer is connected with the neurons in C5.

The output is the last layer of the LeNet-5. Because Lenet-5 is a 10-classification model, there are 10 neuron nodes in this layer. Since this paper implements a task of 2 classifications (living and nonliving), and the sample data is more complex than handwritten datasets, it is obviously unreasonable to use the LeNet-5 to classify living pictures. Thus, referring to the structure and idea of LeNet-5, this paper makes further modificationstoits structure to meet the task requirements. The next section discusses the improvement made forLeNet-5 in this paper.

C. Improved LeNet-5

The recognition accuracy and real-time performance of the model are important indicators to evaluate whether the system is successful. Therefore, this paper improves the LeNet-5 in order to obtain optimal prediction accuracy on the premise of ensuring the high real-time performance of the proposed model. The structure of the in vivo detection model is determined and named LN_Liveness_Detection, which is mainly optimized as follows:

1. Change the size of the partial convolution kernel.

The common convolution kernel size is 7×7,5×5, and 3×3, among others. The convolution kernel of 7 extracts more image details; that is, the larger the convolution kernel, the larger the receptive field and the more features extracted, which is conducive to improving the accuracy of image classification. Because the "face" target in the image is much larger than the "digital" target and the features are complex, in order not to lose too much image information in the early stage, this paper sets the core size of the first and second convolution layers to 7 to increase the receptive field and expand the extraction details. The core size of the following convolution layers remains unchanged, which is still 5.

2. Increase the number of convolution kernels.

Compared with the handwritten data set, the living detection data set is slightly more complex, and the use of fewer convolution cores cannot express the image information, and the use of more convolution cores will increase the amount of calculation but help to improve the accuracy of image classification. Based on this consideration, this paper initially selects 32 convolution cores in the first layer of the model. First, the model parameters will not increase too much; Second, it can extract more low-level information and retain more detailed features.

3. Deepen the model structure.

Due to its low complexity, the shallow neural network cannot better extract the main features in the image, which makes the model classification effect poor, and ultimately affects the recognition rate of the model. In this paper, the number of structural layers of the LeNet-5 is modified, and the original three-layer convolution is increased to five layers to improve the fitting ability of the network.

4. Gap replaces the full connection layer.

Most parameters of CNN are concentrated in the full connection layer. If the traditional full connection layer is adopted, it is bound to increase the model parameters. While the gap belongs to the nonparametric structure, the parameters of the whole model can be reduced, and the prediction speed of the later stage of the model can be improved. In addition, it can effectively prevent overfitting.

Fig. 3 shows the LeNet_Liveness structural diagram for detection, and Table 1 shows the relevant structural parameters of the model. The input layer is 128 × 128 three-channel face pictures. C1, C2, C3, C4, and C5 are convolution layers, and the moving step size is 1. The convolution kernel size of the C1 and C2 layers is 7 × 7. The convolution kernel size of the C3, C4, and C5 layers is 5 × 5. In addition, C1 outputs 32 feature maps, C2 outputs 64 feature maps, C3 outputs 128 feature maps, C4 outputs 256 feature maps, and C5 outputs 512 feature maps. P1, P2, P3, P4, and P5 are all maximum pooling layers. The characteristic diagram of each input is 2 × 2, with a step size of 2, so as to reduce the length and width of the image and reduce the amount of calculation. The gap is the global average pool layer, and the output is the output layer.

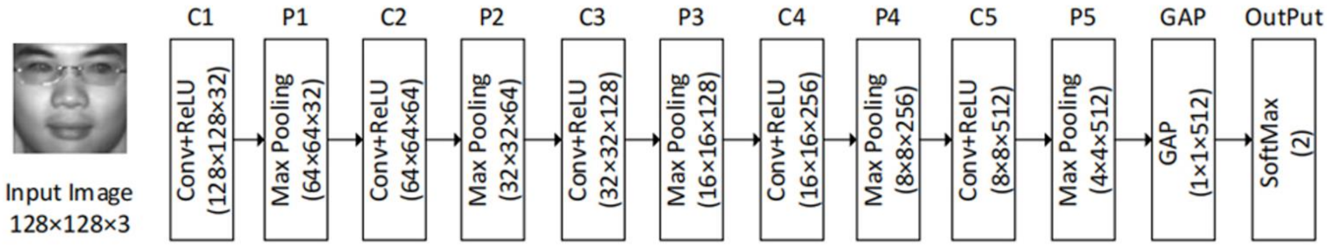


Fig. 3. LeNet_Liveness structure diagram

TABLE I.
MODEL STRUCTURE PARAMETERS

Layer Name	Layer Tyoe	Output Size/Strides	Kernel Size
Input	Input Layer	128x128x3/-	-
C1	Convolution	128x128x32/1	7
P1	Max Pooling	64x64x32/1	2
C2	Convolution	64x64x64/1	7
P2	Max Pooling	32x32x64/2	2
C3	Convolution	32x32x128/1	5
P3	Max Pooling	16x16x128/1	2
C4	Convolution	16x16x256/1	5
P4	Max Pooling	8x8x256/2	2
C5	Convolution	8x8x512/1	1
P5	Max Pooling	4x4x512/2	2
GAP	GAP	1x1x512/1	4
Softmax	Softmax	2/-	-

D. Dataset

The dataset used in the experiment includes the living and nonliving samples. The living image data is thePolyU-NIRFD [10], a public dataset collected by Hong Kong Polytechnic University in 2010. The data set contains near-infrared images of 350 volunteers under different light intensities, different head postures and different expressions. About 100 images were collected for each person, with a total of 38,981 images. The face area was intercepted by a multi-task progressive neural network (MTCNN) face detector [11]. Relevant examples of the samples are shown in Fig.4.The nonliving sample image data is the self-built data set in this paper. This paper randomly selects about 500 images with different head postures and different expression states from the CelebA dataset and uses an HP printer (deskjet 2600) to print each image on A4 size paper to build the attack samples. The relevant examples of theprinted images are shown in Fig.5.

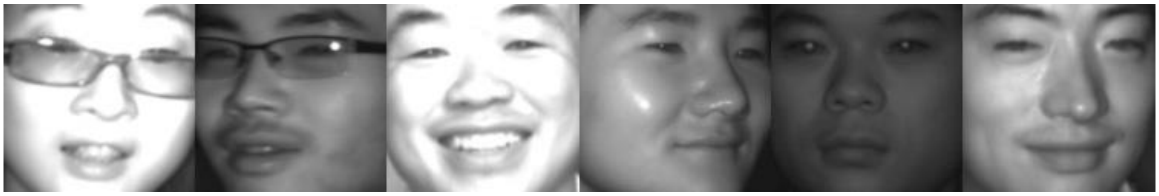


Fig. 4. Liveness face sample

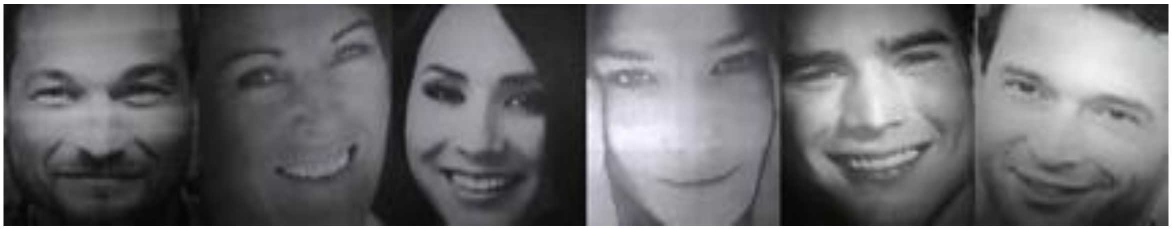


Fig. 5. Sample of printed face images

E. Experimental Environment and Evaluation

The hardware used in the experiment includesan Intel Core i5-8300h processor, NVIDIA GeForce GTX1050TI (4GB video memory) with the memory ofan 8GB computer.The deep learning framework used is TensorFlow-1.13.1, while the image processing library is OpenCV-4.1.0.The Python-3.6.7 programming language is also instrumental in this study.

Moreover, the classification model usually uses accuracy as the evaluation standard of its performance, and the relevant calculation formula is as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (1)$$

The "TP" refers to the number of samples predicted to be positive in the positive sample. The "TN" indicates the number of samples predicted to be negative in the negative sample. The "FP" indicates the number of samples predicted to be positive in negative samples, while the "FN" indicates the number of samples predicted to be negative in the positive sample. The closer the "accuracy" is to 1, the better the classification performance of the model.

IV. Result And Discussion

A. Model Training

In order to better verify the recognition rate of the model, this paper adopts the 10-fold cross-validation method, and the images in each sample set adopt the random sampling method.

During the model training, the batchsize is set to 64. There is a need to train 10 epochs in total and the initial learning rate is 10^{-4}. Multiply the learning rate of every 5 epochs by 0.1.The gradient descent optimizer used is Adam [12], and the three-channel face image is set to 128 × 128 size.Moreover, there is a need to normalize the pixel value to the [0,1] interval using the minimum and maximum normalization method using Equation (2):

$$norm = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \qquad (2)$$

Where the X is the current pixel value, and theX_{min} and X_{max} are the minimum and maximum values of image pixels, respectively.

The simulation results of the 10-fold cross-validation technique are shown in Table II, where the accuracy of 10 groups of test sets is more than 99.90%, and the average accuracy reaches 99.95%. The accuracy difference of each group is small, which denotes that the model has tended to be stable when the epoch reaches 10. Fig.6 shows the iterative process of the first group of experiments. It can be seen from the figure that when the epoch is greater than 4, the accuracy basically remains in a straight line, and the loss value also drops below 0.02. With the increase in the number of iterations, the loss value continues to decline. When the epoch is greater than 5, the loss value basically maintains a straight line and is very close to 0, and finally reaches the steady state.

TABLE II.
10-FOLD CROSS-VALIDATION RESULTS

Category	Test Dataset									
	1	2	3	4	5	6	7	8	9	10
Liveness	99.97	99.97	99.97	100	100	99.97	99.97	99.97	100	99.95
Non-liveness	99.98	99.96	99.91	99.98	99.98	99.93	99.86	99.93	99.91	99.83
Overall	99.96	99.95	99.94	99.95	99.98	99.95	99.91	99.95	99.96	99.90

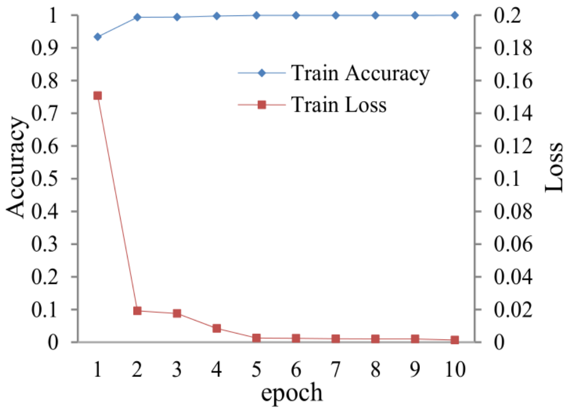


Fig. 6. Data training process

B. *Model Comparison and Result Analysis*

In order to determine the effectiveness of the LeNet modelproposed in this paper, under the condition that the basic conditions such as data preprocessing and dataset division remain unchanged, the modified LeNet, SVM and LeNet-5 are used to classify and identify the living detection datasets. The final experimental test results are shown in Table III.

TABLE III.
INDEXED SIMULATION RESULTS

Algorithm	Accuracy (%)	Average Prediction Time for a Single Picture	
		GPU	CPU
SVM	96.67%	-	4.43
LeNet-5	98.23%	2.03	7.57
Modified LeNet	99.95%	10.77	31.08

It can be seen from Table III that the proposed model obtained the highest accuracy when compared with LeNet-5 and SVM. The average accuracy of SVM is 96.67% while the LeNet-5 obtained a 98.23% accuracy. It is because the characteristics of real people/photos in near-infrared imaging are obvious and easy to distinguish. The experimental result of the LeNet_Livenessmodel proposed in this paper obtained the highest accuracy with 99.95%. The average prediction time of a single picture is 31.08ms on CPU and only 10.77ms on GPU. Compared with SVM and LeNet-5, the speed is slow. This is due to the disadvantages brought by the deepening of model structure and the increase of the number of convolution cores, which reduces the speed.

V. CONCLUSION

In this paper, a living detection method based on improved LeNet-5 is proposed. Based on the structure of LeNet-5, a LeNet-5_Liveness is constructed, and several experimental analyses are carried out for this structure. Finally, the experimental results show that the proposed LeNet_Livenessmodel has high classification accuracy and is very effective in combating nonliving attacks.

REFERENCES

[1] Q. Xiao, “Security issues in biometric authentication,” Proc. from 6th Annu. IEEE Syst. Man Cybern. Inf. Assur. Work. SMC 2005, vol. 2005, pp. 8–13, 2005, doi: 10.1109/IAW.2005.1495927.

[2] B. H. Rudall, “Biometric technology,” Kybernetes, vol. 30, no. 2, pp. 407–421, 2001, doi: 10.1108/k.2001.06730baa.004.

[3] Emilio Mordini and Dimitros Tzovaras, Second Generation Biometrics: The Ethical, Legal and Social Context. 2012.

[4] N. Anot and K. K. Singh., “A Review on Biometrics and Face Recognition Techniques.,” Int. J. Adv. Res., vol. 4, no. 5, pp. 783–786, 2016, doi: 10.21474/ijar01/522.

[5] P. Cai and H. min Quan, “Face anti-spoofing algorithm combined with CNN and brightness equalization,” J. Cent. South Univ., vol. 28, no. 1, pp. 194–204, 2021, doi: 10.1007/s11771-021-4596-y.

[6] G. Chenqiang, L. Xindou, Z. Fengshun, and M. Song, “Face Liveness Detection Based on Improved CNN with Context and Texture Information.” Chin. J. Electron., vol. 28, no. 6, pp. 1092-1098, 2019, doi: https://doi.org/10.1049/cje.2019.07.012.

[7] C. -C. Jay Kuo, “Understanding Convolutional Neural Networks with a Mathematical Model,” J. Vis. Commun. Image R., 2016, doi: 10.1016/j.jvcir.2016.11.003

[8] M. Chihaoui, A. Elkefi, W. Bellil, and C. Ben Amar, “A survey of 2D face recognition techniques,” Computers, vol. 5, no. 4, pp. 1–28, 2016, doi: 10.3390/computers5040021.

[9] K. Delac and M. Grgic, “A survey of biometric recognition methods,” Proc. Elmar - Int. Symp. Electron. Mar., no. June, pp. 184–193, 2004, doi: 10.1109/ELMAR.2004.1356372.

[10] K. Solanki and P. Pittalia, “Review of Face Recognition Techniques,” Int. J. Comput. Appl., vol. 133, no. 12, pp. 20–24, 2016, doi: 10.5120/ijca2016907994.

[11] S. F. Kak, F. Mahmood Mustafa, and P. Valente, “A Review of Person Recognition Based on Face Model,” vol. 2018, pp. 1–19, 2018, doi: 10.23918/iec2018.01.

[12] B. Achermann and H. Bunke, “Combination of Classifiers on the Decision Level for Face Recognition,” Tech. Bericht IAM-96-002, Inst. für Inform. Univ. Bern, Schweiz, no. January, 1996.

[13] X. Li, W. Wu, T. Li, Y. Su, and L. Yang, “Face Liveness Detection Based on Parallel CNN,” J. Phys. Conf. Ser., vol. 1549, no. 4, 2020, doi: 10.1088/1742-6596/1549/4/042069.

[14] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, “On the Detection of Digital Face Manipulation,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 5780–5789, 2020, doi: 10.1109/CVPR42600.2020.00582.

[15] N. K. Ratha, J. H. Connell, and R. M. Bolle, “An analysis of minutiae matching strength,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 2091 LNCS, pp. 223–228, 2001, doi: 10.1007/3-540-45344-x_32.

[16] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “Deepfakes and beyond: A Survey of face manipulation and fake detection,” Inf. Fusion, vol. 64, pp. 131–148, 2020, doi: 10.1016/j.inffus.2020.06.014.

[17] L. Liu, O. De Vel, Q. L. Han, J. Zhang, and Y. Xiang, “Detecting and Preventing Cyber Insider Threats: A Survey,” IEEE Commun. Surv. Tutorials, vol. 20, no. 2, pp. 1397–1418, 2018, doi: 10.1109/COMST.2018.2800740.

[18] W. Y. X IAO Bing, “Survey ofHuman Face Recognition,” 2005.

[19] X. Gao, L. Liu, and X. Zhu, “Research on the main threat and prevention technology of computer network security,” IOP Conf. Ser. Earth Environ. Sci., vol. 632, no. 5, 2021, doi: 10.1088/1755-1315/632/5/052065.

[20] Y. Li, R. Liu, X. Liu, H. Li, and Q. Sun, “Research on Information Security Risk Analysis and Prevention Technology of Network Communication Based on Cloud Computing Algorithm,” J. Phys. Conf. Ser., vol. 1982, no. 1, 2021, doi: 10.1088/1742-6596/1982/1/012129.

[21] P. P. P. Linn and E. C. Htoon, “Face Anti-spoofing using Eyes Movement and CNN-based Liveness Detection,” 2019 Int. Conf. Adv. Inf. Technol. ICAIT 2019, pp. 149–154, 2019, doi: 10.1109/AITC.2019.8921091.