Information Dynamics and Applications
https://www.acadlore.com/journals/IDA

Advancements in Image Recognition: A Siamese Network Approach

Jiaqi Du[1, 2*], Wanshu Fu[2, 3], Yi Zhang[4], Ziqi Wang[5]

[1] Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing, 100083 Beijing, China
[2] School of Management and Engineering, Capital University of Economics and Business, 100070 Beijing, China
[3] School of Information, Central University of Finance and Economics, 100081 Beijing, China
[4] School of Environment, Education and Development, The University of Manchester, M13 9PL Manchester, UK
[5] Faculty of Engineering and Information Technology, The University of Melbourne, 1446535 Melbourne, Australia

[*] Correspondence: Jiaqi Du (djqo30923@cueb.edu.cn)

Received: 04-05-2024 **Revised:** 05-20-2024 **Accepted:** 05-31-2024

Citation: J. Q. Du, W. S. Fu, Y. Zhang, and Z. Q. Wang, "Advancements in image recognition: A siamese network approach," *Inf. Dyn. Appl.*, vol. 3, no. 2, pp. 89–103, 2024. https://doi.org/10.56578/ida030202.

Abstract: In the realm of computer vision, image recognition serves as a pivotal task with extensive applications in intelligent security, autonomous driving, and robotics. Traditional methodologies for image recognition often grapple with computational inefficiencies and diminished accuracy in complex scenarios and extensive datasets. To address these challenges, an algorithm utilizing a siamese network architecture has been developed. This architecture leverages dual interconnected neural network submodules for the efficient extraction and comparison of image features. The effectiveness of this siamese network-based algorithm is demonstrated through its application to various benchmark datasets, where it consistently outperforms conventional approaches in terms of accuracy and processing speed. By employing weight-sharing techniques and optimizing neural network pathways, the proposed algorithm enhances the robustness and efficiency of image recognition tasks. The advancements presented in this study not only contribute to the theoretical understanding but also offer practical solutions, underscoring the significant potential and applicability of siamese networks in advancing image recognition technologies.

Keywords: Image recognition; Deep learning; Siamese networks; Weight sharing

1 Introduction

As an important branch in the field of computer vision, image recognition has always been a hot and difficult research topic. With the vigorous development of deep learning technology, especially the wide application of Convolutional Neural Networks (CNNs) in image recognition tasks, traditional image recognition methods are gradually being replaced by deep learning algorithms. However, in the face of complex and changeable image scenes and large-scale datasets, traditional deep learning models still face problems such as large computational costs and poor generalization performance [1]. As a special neural network structure, the siamese networks can effectively extract the features of the input data and calculate the similarity by sharing weights, showing unique advantages in image recognition tasks.

This research aims to study the image recognition algorithm based on siamese networks, and improve the recognition performance and computational efficiency of the algorithm by optimizing the network structure and improving the training strategy. At the same time, this study also explores the applicability of image recognition algorithms based on siamese networks in different application scenarios, aiming to provide new ideas and methods for the development of the image recognition field.

The remaining structure of this study is as follows: Section 2 is a literature review in the field related to this study; Section 3 is research methods and data preprocessing; Section 4 is model construction; Section 5 is result analysis; and Section 6 is the conclusion and future research prospect of this study.

2 Literature Review

At first, researchers in the field of image recognition mainly relied on image recognition methods based on statistical features. For example, Zhou et al. [2] introduced the extraction method of invariant features at the local

scale of images, and proposed a hybrid multi-scale representation method using pyramids and scale space to improve the real-time performance of image recognition. Shi and Zhang [3] proposed a method to realize moving target detection using a single Synthetic Aperture Radar (SAR) image, and proved the effectiveness of the proposed method with experiments. Galić et al. [4] introduced the use of different machine learning algorithms for image recognition, including feature extraction, feature selection, classifier design, etc. In general, the image recognition method based on statistical features involves statistical analysis of image rules, the extraction of features that reflect the essence of images, and the establishment of recognition models based on decision theory. However, this method has limitations, such as ignoring the spatial structure relationship of images, and the number of features surges, leading to difficulty in extraction and classification, especially for images with obvious structural features. The statistical recognition effect is not good.

With the rise of deep learning technology, the performance of image recognition has greatly improved. Chen [5] verified the effectiveness of CNNs in image recognition tasks through experiments. A variety of model optimization strategies were discussed to further improve the performance of CNNs. Qin et al. [6] used the CNN architecture to design a flower image classification and recognition model based on deep learning, and verified the effectiveness of the designed flower image classification and recognition model through experiments. The key to the image recognition method based on deep learning is to automatically extract the image features, and make the model recognize and classify the image through training and optimization. However, despite the great success of deep learning-based image recognition methods in the field of image recognition, there are still some challenges [7]. For example, deep learning methods often rely on large amounts of labeled data for model training. However, in practice, the sample size of some classes can be very small, making it difficult for deep learning models to fully learn their features. Therefore, the small-shot learning problem has become an important challenge in the field of deep learning image recognition. In addition, the generalization performance of deep learning models is also a challenge. Due to complex structures with a large number of parameters, deep learning models tend to be prone to overfitting, i.e., they perform well on training data but poorly on test or new data [8].

In order to solve the above problems, researchers have begun to explore new network structures and learning methods. Among them, the siamese networks have been widely used and explored because of their unique network structure. For example, Valero-Mas et al. [9] outlined the application of siamese networks in image classification tasks with a small number of samples, emphasized the importance of ensemble and feature learning in improving classification performance, and compared the advantages of siamese networks with other methods. He [10] compared the traditional Visual Geometry Group Network (VGGNet) series model with the siamese VGGNet model. The siamese VGGNet model made full use of the correlation and difference between image pairs, confirming its effectiveness in enhancing accuracy and robustness. Ren et al. [11] combined the siamese CNN with Region Proposal Network (RPN) to solve the problem of untimely network model updates and insufficient training datasets during online tracking. Siamese networks learn similarity measures between image pairs by sharing weights, which gives them a significant advantage when dealing with tasks such as small-shot learning and fine-grained image recognition. In addition, the siamese networks took full advantage of the correlation and difference between image pairs, improving the accuracy and robustness of recognition.

However, this study of image recognition based on siamese networks is still in the development stage, and there are still many problems and challenges to be solved and explored. First of all, due to their special network structure, the siamese networks require more computing resources and time for training and optimization. Secondly, the siamese networks have high requirements for the quality and preprocessing of the input image pairs; otherwise, their performance may be affected. In summary, as a new type of network structure and learning method, siamese networks have a wide range of application prospects in the field of image recognition. However, they still have some disadvantages and shortcomings, which need to be further explored and optimized. Future research could focus on how to improve the training efficiency, generalization ability, and performance of the siamese networks in practical applications.

3 Research Methods and Data Preprocessing

3.1 Siamese Networks

Figure 1 shows the composition of the siamese neural networks. Siamese networks have a special neural network structure, which mainly consists of two or more identical subnet modules, which share the same weights and parameters. Each subnetwork receives an input sample and generates a representation vector. These vectors are then used to calculate the similarity between the input samples. The weight-sharing mechanism between subnets is mainly reflected in the backpropagation phase. When the weights of one subnet are updated during the training process, these updates are immediately reflected in the other subnet. This process of synchronous updates ensures that both subnetworks use the same set of weights when processing different input data.

The siamese networks' comparison module, which comes after the subnet, receives the eigenvectors from the subnet as input and compares their similarity to determine the relationship between the input samples. The

comparison module employs several methods to calculate the similarity between eigenvectors. Common methods include Euclidean distance, cosine similarity, inner product, etc. In addition, more complex neural network layers or functions can be used to calculate the similarity or difference between feature vectors, such as multilayer perceptrons (MLPs), CNNs, etc.

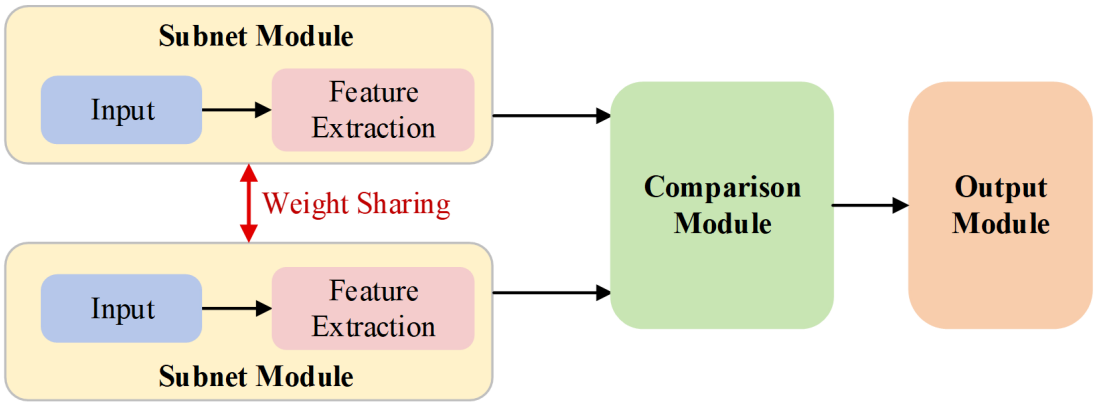


Figure 1. Structure of the siamese networks

Finally, the output module of the siamese networks is responsible for translating the output of the comparison module into the final decision-making result. The output module can also be customized to meet the needs of the specific task. For example, in a target tracking task, the output module may output the target location and size. In a face recognition task, the output module may output the identity information of the face, and so on.

Structurally, conventional neural networks typically consist of only one network that processes a single input sample and outputs the corresponding predictions. The siamese networks consist of two or more subnets that share weights and parameters, which are suitable for tasks that require the similarity comparison of two input samples. Functionally, the main role of the siamese networks is to calculate the similarity between two input samples, while conventional neural networks focus more on the prediction and classification of a single input sample. In addition, because of their special structure, the siamese networks are also suitable for small- or single-shot learning. That is, they can maintain good performance even when the training data is limited.

Siamese networks have a wide range of applications in the field of computer vision, especially in tasks involving comparing the similarity of two input samples. The following is a brief introduction to some of their application scenarios in computer vision:

(a) Face recognition [12]: In the face recognition task, by inputting a pair of face images, the siamese networks can learn to extract the image features, and judge whether the two images are the same person's face by calculating the similarity between the features. This method is particularly effective for dealing with face images with different angles, lighting conditions, and variations in expression [13].

(b) Image retrieval: In the image retrieval task, given a query image, the system can find images similar to it in a large image database. By learning the feature representation of the image, the siamese networks can accurately calculate the similarity between the query image and the image in the database, thereby selecting the image with the highest similarity to return.

(c) Target tracking [14]: In the target tracking task, the siamese networks can identify and continuously track the targets in the video sequence. By comparing the targets in the current and previous frames, the networks can learn how the object appears and predict its position in subsequent frames. This approach is very effective for dealing with challenging problems such as complex backgrounds, occlusions, and target deformations.

3.2 Data Preprocessing

In this study, three sets of public datasets were used: MNIST, Fashion-MNIST and CIFAR10.

Among them, MNIST and Fashion-MNIST have the same data format and scale. Both datasets contain 60,000 training samples and 10,000 test samples. Each sample is rendered in a 28x28 pixel grid with a single grayscale channel. As shown in Table 1, the MNIST is a classic handwritten digital image dataset, with ten categories from 0 to 9 corresponding to handwritten digital images from 0 to 9. However, ten categories from 0 to 9 in the Fashion-MNIST dataset correspond to different kinds of clothing. This is the main difference between those two datasets. Compared with MNIST, the image content of Fashion-MNIST is more diverse, complex, and challenging. In this experiment, 60,000 original training samples were divided into 48,000 training samples and 12,000 validation samples.

The CIFAR10 dataset contains ten types of objects, with label values ranging from 0 to 9. The CIFAR10 dataset consists of 50,000 training samples and 10,000 test samples, each of which is a 32x32-pixel RGB image. In

this experiment, 50,000 original training samples were divided into 40,000 training samples and 10,000 validation samples. During preprocessing, the image was normalized to the range of [0,1] and a channel dimension was added.

Table 1. Dataset categories and object correspondence

Categories \ Datasets	MNIST	Fashion-MNIST	CIFAR10
0	0	T-shirt/top	Plane
1	1	Trouser	Car
2	2	Pullover	Bird
3	3	Dress	Cat
4	4	Coat	Deer
5	5	Sandals	Dog
6	6	Shirt	Frog
7	7	Sneaker	Horse
8	8	Bag	Boat
9	9	Ankle boots	Truck

Figure 2. Visualization results of (a) MNIST; (b) Fashion-MNIST; and (c) CIFAR10 image pairs

The siamese networks built in this experiment have two shared-weight subnets, providing inputs for two network models. This leads to the concept of positive and negative pairs. Positive pairs refer to images in the same category, while negative ones refer to images in different categories. In order to visually verify whether the image pair generation process in this experiment works normally, the generated image pair was randomly selected and visualized once. Subgraph (a), Subgraph (b) and Subgraph (c) of Figure 2 show the visualization. It can be observed

that for each pair of images, the same class is labeled as POS and the different classes are labeled as NEG.

4 Modelling

4.1 Algorithmic Framework

As shown in Figure 3, the two subnets are referred to as subnets A and B. Both subnets share the same structure and weights. Each subnet starts with multiple convolutional layers that extract features from the input image. Convolutional layers filter images through convolutional operations to capture local spatial features. The convolution operation is usually followed by an activation function, such as the ReLU function, to increase the nonlinearity of the network. The pooling layer is used to reduce the dimensionality of the feature map while retaining important feature information. After multiple convolutional and pooling layers, a flattened layer is typically used to flatten the multidimensional feature map into a one-dimensional vector to input into a fully connected layer. The fully connected layer is used to map feature vectors to the sample space, generating a feature representation of each sample. After the feature vectors extracted by the subnet pass through the comparison and output modules, the similarity value between the image pairs is finally output.

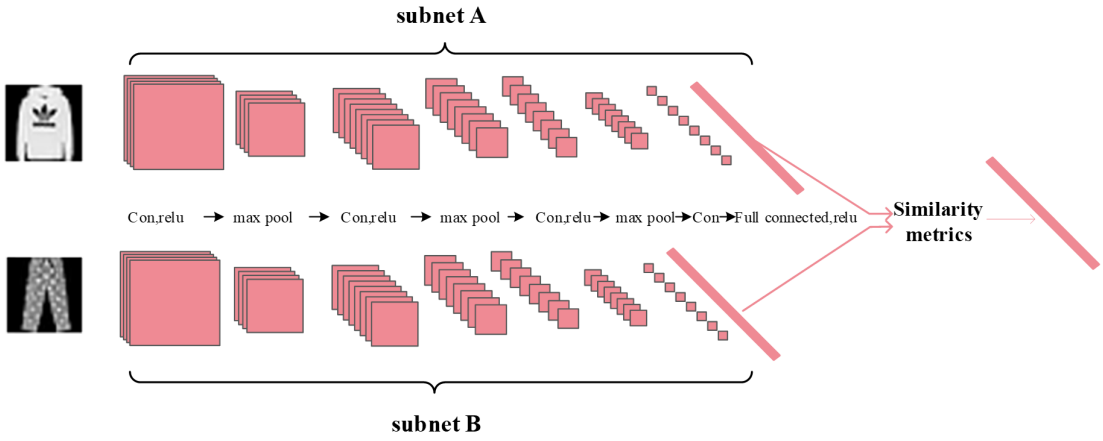


Figure 3. Framework of the siamese networks

4.2 Subnet Module

In this study, the classical CNN structure LeNet-5 was used as the subnet of the siamese networks. LeNet-5 is an earlier CNN structure proposed by Yann LeCun et al. in 1998 for processing smaller images. Figure 4 shows the basic structure of LeNet-5 as siamese network subnets.

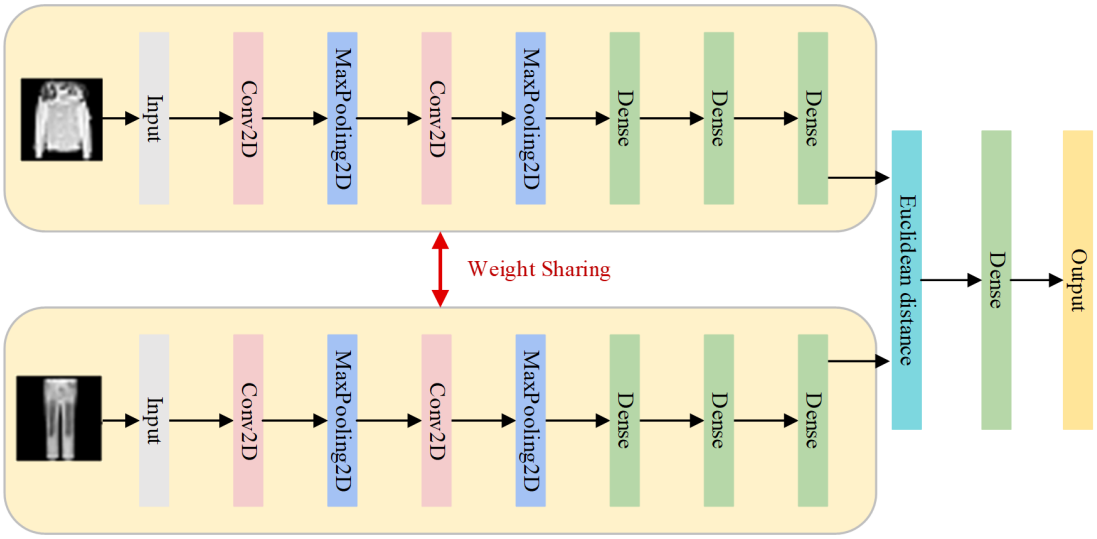


Figure 4. Siamese networks based on the LeNet-5 subnet

4.3 Comparison Module

The similarity metric function of the comparison module is used to accurately calculate the similarity between the eigenvector outputs of the two subnetworks. This module measures the similarity or distance between vectors through a specific algorithm or measurement method. Common similarity measures include Euclidean distance, cosine similarity, etc. In this study, Euclidean distance serves as a comparison module, and the formula for calculation is as follows:

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

where, x_i is the i-th coordinate of the eigenvector x, with $i = 1, 2 \ldots n$; y_i is the i-th coordinate of the eigenvector y, with $i = 1, 2 \ldots n$; and $d(x, y)$ is the distance between the two eigenvectors x and y.

4.4 Output Module

In most scenarios, the output module tends to be a simple classifier, such as a logistic regression or softmax layer. When the siamese networks are used for image recognition or verification tasks, the output module outputs the probability that two input images belong to the same category. In this study, the sigmoid function was used as an output module to output the similarity values of the two images.

4.5 Training Module

In the training module of the siamese networks, the loss function and optimizer work together to drive the learning and optimization processes of the network. The loss function is responsible for quantifying the difference between the model prediction and the actual label, providing a clear optimization goal for network training. According to the loss function gradient, the optimizer updates the network parameters through a certain algorithm, thereby gradually reducing the loss value and improving the prediction performance of the model. This process ensures that the model can gradually approach the optimal state during the training process, and improves the ability to judge the similarity between samples.

Table 2. Adam update rules

	Adam Update Rules	
(a)	Calculation of the gradient for the t time step	$\mathrm{g_t} = \nabla_\theta J\left(\theta_{t-1}\right)$
(b)	Calculation of the exponential moving average of the gradient	$\mathrm{m_t} = \beta_1 m_{t-1} + \left(1 - \beta_1\right) g_t$
(c)	Calculation of the exponential moving average of the gradient squared	$v_\mathrm{t} = \beta_2 v_{t-1} + \left(1 - \beta_2\right) g_t^2$
(d)	A deviation correction is made for $\mathrm{m_t}$	$\hat{\mathrm{m}}_\mathrm{t} = m_t / \left(1 - \beta_1^t\right)$
(e)	A deviation correction is made for v_t	$\hat{v}_\mathrm{t} = v_t / \left(1 - \beta_2^t\right)$
(f)	Update of the parameters	$\theta_\mathrm{t} = \theta_{t-1} - \alpha * \hat{\mathrm{m}}_\mathrm{t} / (\sqrt{\hat{v}} + \varepsilon)$

(a) Loss function

The design of the loss function is usually related to the specific task. In the classification task of the siamese networks, binary cross-entropy or contrastive loss functions are usually used. In this study, a contrastive loss function was used to encourage the networks to produce similar and distant feature representations for similar and dissimilar samples, respectively. The formula for calculation is as follows:

$$Loss = \frac{1}{N} \sum_{i=1}^{N} (1 - y_i) \cdot d^2 + y_i \cdot \max(margin - d, 0)^2 \tag{2}$$

where, N represents the number of samples; y_i represents the label of the i-th sample pair, which is 0 or 1 ; d represents the sample similarity value predicted by the networks; and *margin* is the given hyperparameter. A value of 1 was taken. This indicates that when d is greater than or equal to 1, the sample pairs are very similar, and the loss value is 0. When $y_i = 1$ (i.e., the samples belong to the same class), the loss function is only $\sum y_i \cdot \max(margin - d, 0)^2$. When d is smaller (i.e., the predicted similarity value is smaller), the loss function is larger. When $y_i = 0$ (i.e., the samples belong to different classes), the loss function is only $\sum (1 - y_i) \cdot d^2$. When d is larger (i.e., the predicted similarity value is smaller), the loss function is larger.

(b) Optimizer

The Adaptive Moment Estimation (Adam) optimizer, which combines the advantages of AdaGrad and RMSProp optimization algorithms, comprehensively considers the first-order (i.e., the mean of the gradient) and the second-order (i.e., the uncentralized variance of the gradient) moment estimations of the gradient to calculate the update step size [15].

Table 2 lists the Adam update rules, where g_t represents the gradient of time t; β_1 and β_2 represent the exponential decay rate, with the default $\beta_1 = 0.9$ and $\beta_2 = 0.999$; m_t and ν_t represent the weighted average and biased deviation of the gradient, with $m_0 = 0$ and $v_0 = 0$, respectively; $\hat{m}_t$ and $\hat{v}_t$ are the deviation correction of m_t and v_t; θ_t is the parameter at time t; θ_{t+1} is the update parameter at $t+1$; α and ε are the initialization parameters, with the default learning rate $\alpha = 0.001$ and $\varepsilon = 10^{-8}$ to avoid the divisor becoming 0 .

5 Result Analysis

In this study, the experimental environment was configured on a system running the Windows operating system, equipped with an Intel Core i7 processor. The development environment utilized was PyCharm, operating under Python version 3.10. This setup was further enhanced by integrating the TensorFlow deep learning framework, which, in conjunction with CUDA and cuDNN, facilitated accelerated computational performance. These technological integrations were pivotal in establishing an efficient and stable platform for conducting image recognition experiments.

5.1 Results of the Comparative Experiment

Figure 5 shows the results of 50 training sessions on the siamese networks based on LeNet-5 (SN-LeNet-5) of the MNIST dataset. After 50 training sessions, the accuracy of the training, validation and test sets reached 99.95%, 98.83% and 99.11%, respectively.

Figure 5. Iterative plot of MNIST on SN-LeNet-5

Table 3 shows the test performance comparison of the recognition methods based on the K-proximity method, Histogram of Oriented Gradients (HOG) and neural networks on the MNIST handwriting dataset in recent years. The applied SN-LeNet-5 exhibits superior recognition performance compared to other methods, surpassing the Lenet-5 model by 0.06%.

Table 3. Comparison of the test accuracy

Method	Accuracy (%)
Lenet-5 [16]	99.05
HOG_PCA [17]	98.39
HOG_SVM [18]	97.25
SNN_STPD [19]	98.4
CNN [20]	98.99
CNN_SVM [21]	99.10
SN-LeNet-5	99.11

Table 4 shows the time required for the identification of MNIST datasets by the SN-LeNet-5 and the traditional LeNet-5 networks. The time used by the two methods in the table was accumulated after performing a test of 10,000 images. The SN-LeNet-5 demonstrates a faster recognition speed.

Table 4. Comparison of time performance

Method	Test Time (s)
Lenet-5	9.37
SN-LeNet-5	4.04

Figure 6 shows the output results of the SN-LeNet-5 model on the handwritten digital image test set. In the figure, the label directly above each pair of images indicates their similarity values. The closer the similarity value is to 1, the greater the similarity between the pair of images, which may belong to the same category. The closer the similarity value is to 0, the greater the difference between the images, which may belong to different categories. It can be seen from the visualization results that the similarity prediction values of the siamese networks for image pairs in different types and in the same class are very close to 0 and 1, respectively, indicating that the model performs well on the test set.

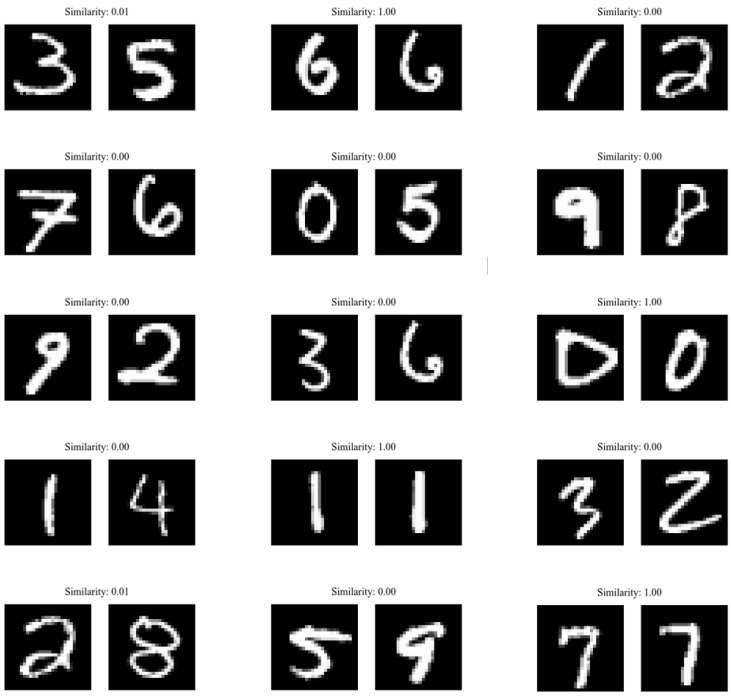


Figure 6. Visualization of the test results of the MNIST dataset

5.2 Analysis of Model Generalization

Figure 7 and Figure 8 show the iterative plots of the siamese network model based on LeNet-5 on the Fashion-MNIST and CIFAR10 datasets, respectively. The accuracy of the Fashion-MNIST on the test set reached 91.65%

after 100 training times on the siamese networks. The CIFAR10 dataset did not reach complete convergence after 1,000 times of training on the networks. Therefore, it was trained 500 times on the basis of 1,000 times. The accuracy of the final model on the test set reached 81.64%.

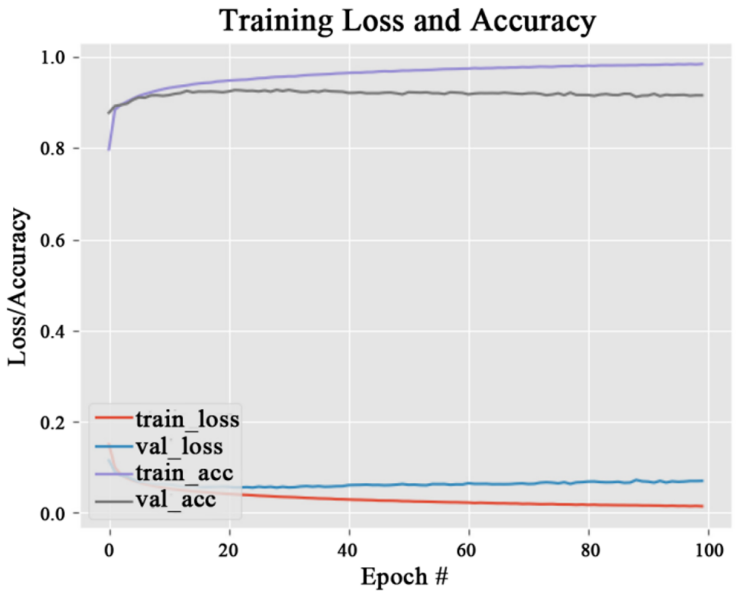


Figure 7. Iterative plot of Fashion-MNIST on SN-LeNet-5

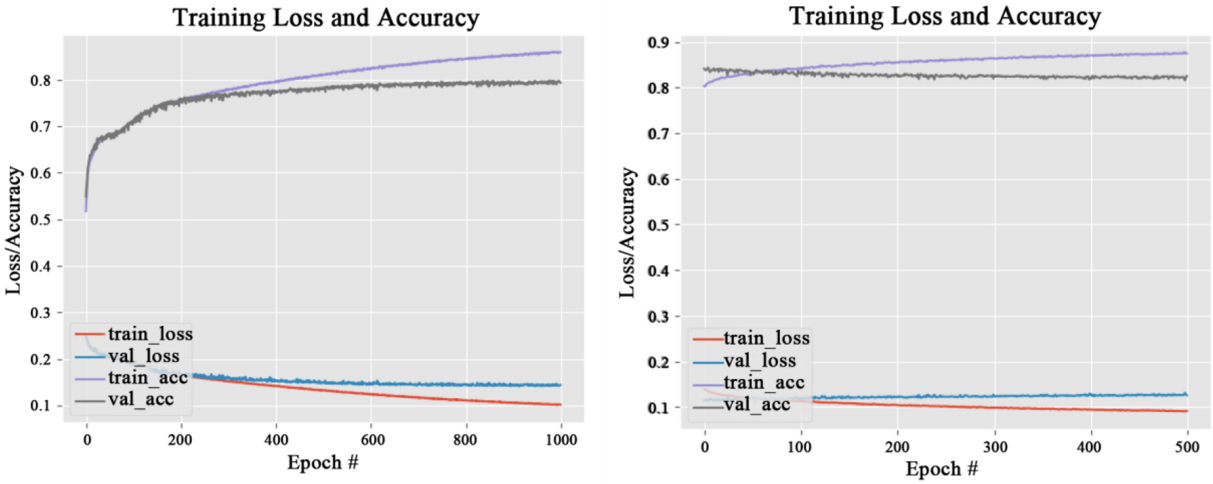


Figure 8. Iterative plots of CIFAR10 on SN-LeNet-5

Table 5 shows the results and running times of the Fashion-MNIST and CIFAR10 datasets on the siamese network model based on LeNet-5. It can be seen from the results that the siamese network model based on LeNet-5 constructed in this experiment has good generalization ability on these two datasets. This means that the model may also have some power to process new or unseen data.

Table 5. Algorithm performance of the siamese networks on Fashion-MNIST and CIFAR10

Datasets	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)	Training Time (s/epoch)	Test Time (s)
Fashion-MNIST	98.56	91.66	91.65	18.73	3.73
CIFAR10	87.46	82.53	81.64	27.32	4.35

Figure 9 and Figure 10 show the output results of the siamese network model based on LeNet-5 for the Fashion-MNIST and CIFAR10 test sets, respectively. According to the visualization results, based on the similarity value of the output image pairs, a high accuracy can be obtained to determine whether the images belong to the same class, showing that the constructed model has good generalization ability on these two datasets.

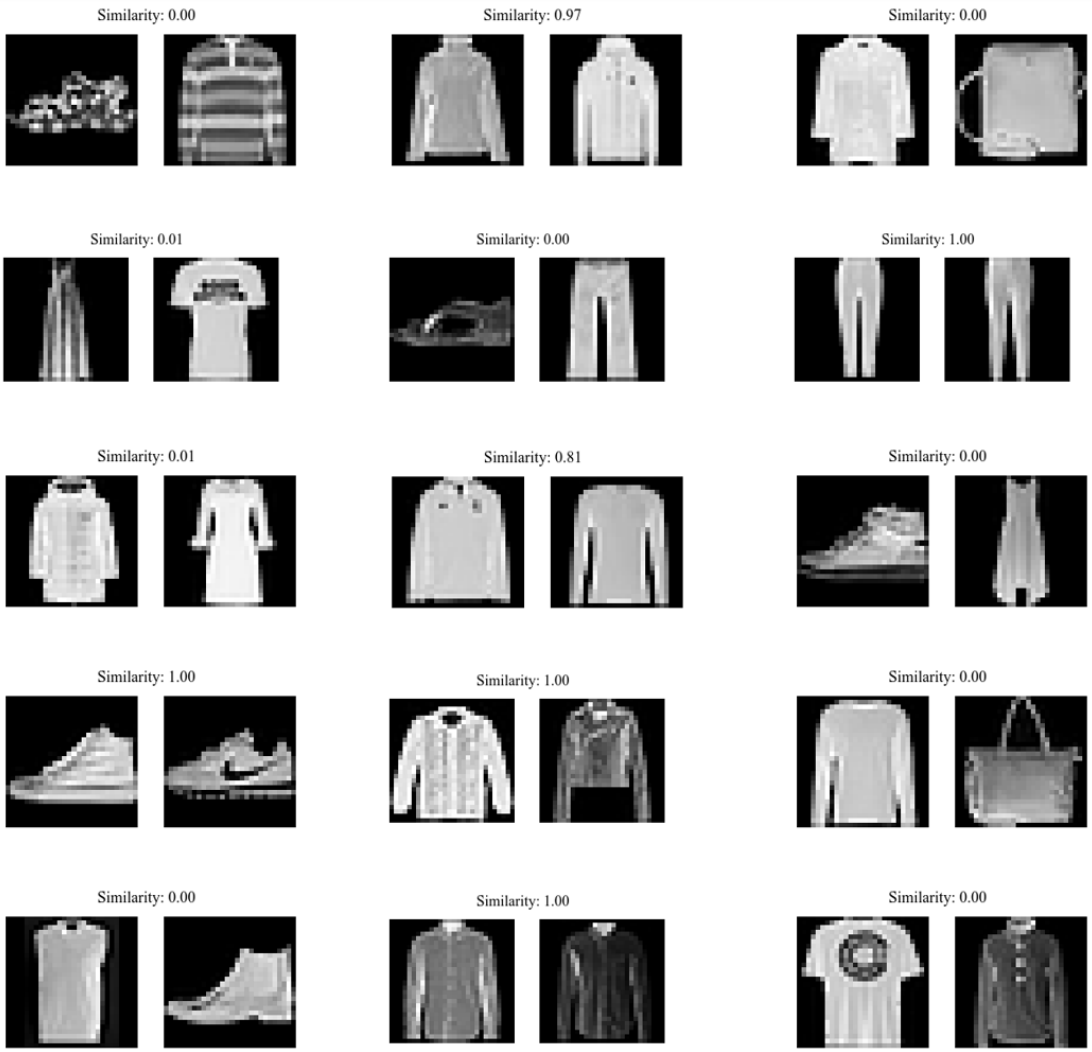


Figure 9. Visualization of the test results of the Fashion-MNIST dataset

5.3 Analysis of Parameter Sensitivity

Parameter sensitivity analysis is a key process to evaluate the sensitivity of model performance to different parameter settings. This chapter delves into the key parameters in the siamese networks and analyzes their impact on model performance, aiming to provide specific guidance for model optimization.

5.3.1 Analysis of learning rates

In this experiment, a siamese network model based on the LeNet-5 architecture was used to train the MNIST dataset. In order to optimize the training process of the model, different learning rate settings were tried, namely, 0.0001, 0.001, and 0.01. During the training, a batch size of 64 was set, meaning that 64 samples were processed each time the weights were updated. The number of iterations was set to 50 rounds to ensure the full convergence of the model. Through this series of settings, it is expected to find the most suitable learning rate to achieve the best model performance.

Table 6. Training accuracy under different learning rates

Learning Rates	0.0001	0.001	0.01
Training Accuracy (%)	99.75	99.95	98.10

Subgraphs (a) and (b) of Figure 11 show the iteration plots with different learning rates. Table 6 shows the accuracy of the training set at each learning rate.

At a learning rate of 0.0001, the convergence speed of the model appears to be relatively slow. However, after a long period of training, it finally achieves a high accuracy rate of 99.75%. Although this result is already quite good,

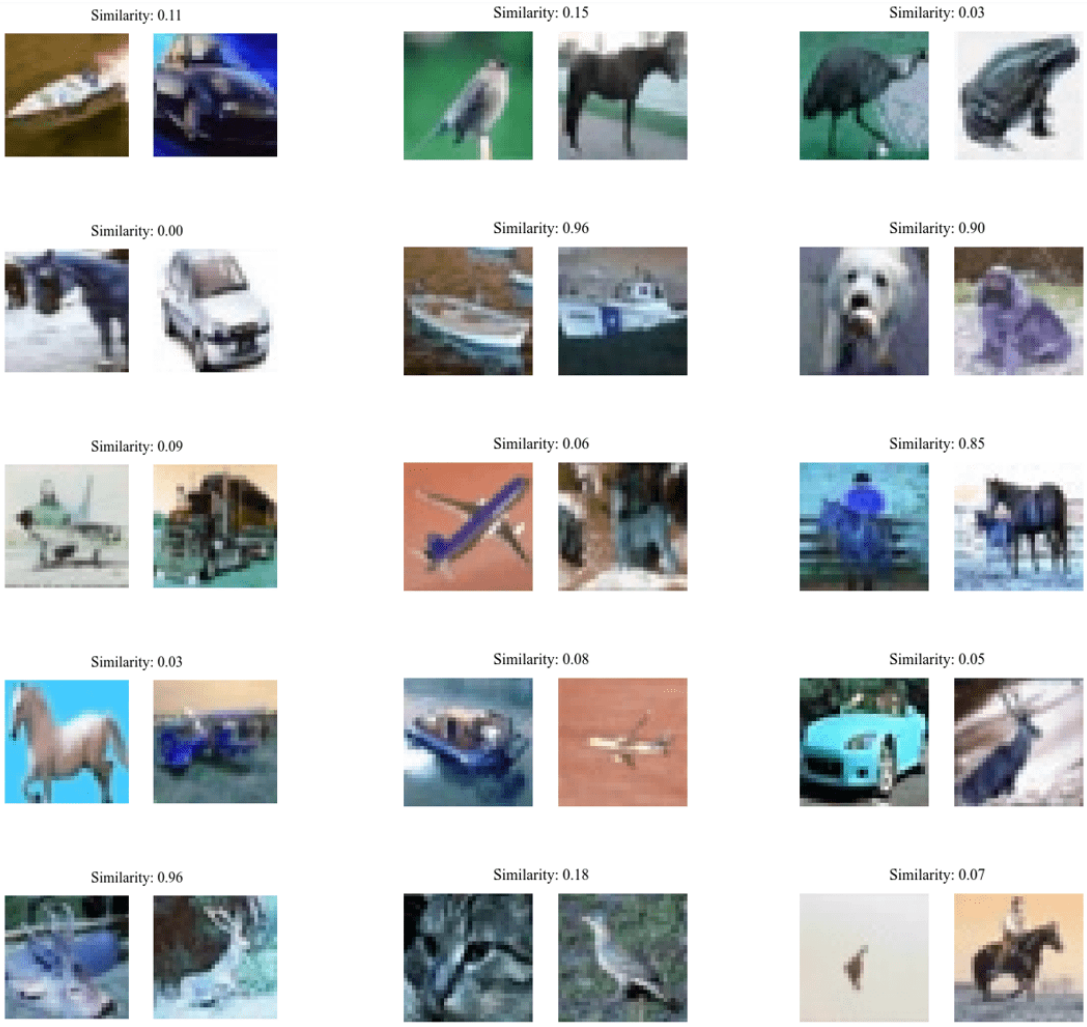


Figure 10. Visualization of the test results of the CIFAR10 dataset

it may be necessary to find a better balance given the time and efficiency of training.

When the learning rate increases to 0.01, the convergence speed of the model significantly accelerates, and the training process becomes faster. However, this setting causes the model to oscillate significantly during the training process, leading to instability when updating the weights. Finally, although the model can converge, its accuracy is relatively low at 98.10%, indicating that it may be more likely to fall into the local optimal solution under the high learning rate, which affects its generalization ability.

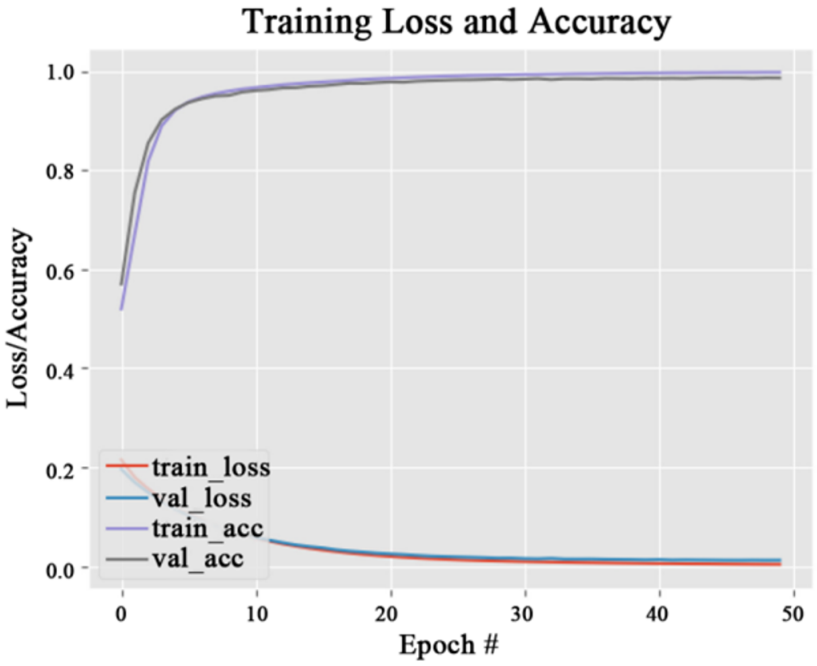
After experimenting with various learning rates, it was found that when the learning rate was set to 0.001, the model reached a good balance between convergence speed and accuracy, as shown in Figure 5. This setting not only keeps the model at a fast convergence speed, but also does not show obvious oscillation during training. In the end, the model achieved the highest accuracy rate of 99.95%, proving that it can achieve excellent performance and accuracy with an appropriate learning rate.

Table 7. The final training accuracy and time under different batch sizes

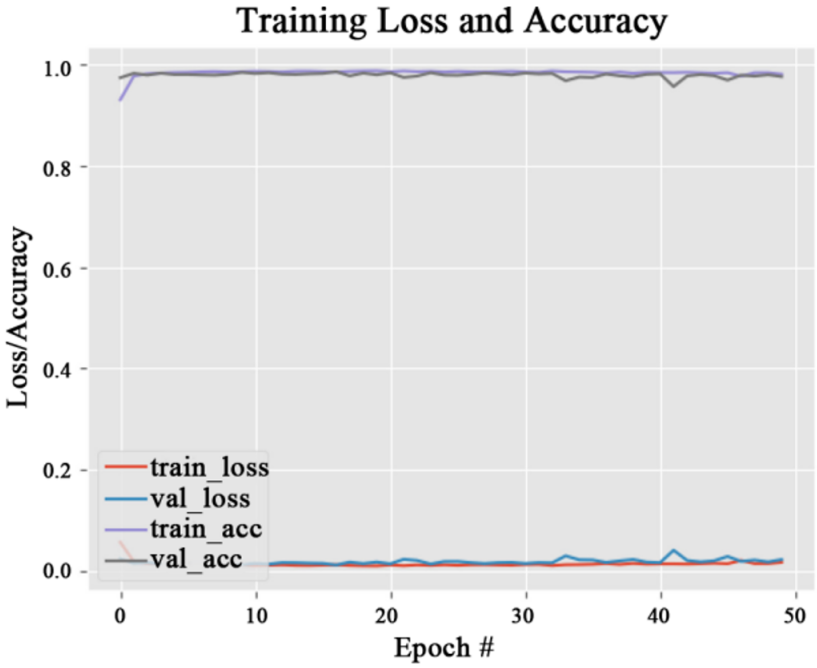
Batch Size	32	64	96
Training Accuracy (%)	99.93	99.95	99.83
Training Time (s/epoch)	34.76	17.36	16.80

5.3.2 Analysis of batch size

In order to analyze the influence of batch size on the training process and model performance during the experiment, three different batch sizes of 32, 64 and 96 were selected and trained under the same experimental conditions. Figure 12 and Table 7 show the experimental results.



(a)



(b)

Figure 11. Iterative plots with learning rates of (a) 0.0001; and (b) 0.01

According to the experimental results, as the batch size increases, the training time per epoch gradually decreases from 34.76 seconds to 16.80 seconds. However, although the training speed further improves when the batch size is 96, the final accuracy of the model decreases slightly to 99.83%. When the batch size is 64, not only is the training speed faster, but the accuracy of the model reaches the highest (99.95%). Furthermore, a comparative analysis of Figure 12 and Figure 5 reveal that the loss values for batch sizes of 32 and 96 converge more slowly compared to a batch size of 64, and fail to reach the minimum loss value. These results demonstrate that the experimental setup can achieve high model accuracy and fast loss convergence while maintaining training speed when the batch size is 64.

(a)

(b)

Figure 12. Iterative plots with batch sizes of (a) 32; and (b) 96

6 Conclusions and Prospects

This study focuses on the image recognition of the siamese networks. After introducing the importance and challenges of image recognition, the relevant research progress was reviewed. Then, the structure and weight-sharing mechanism of the siamese neural networks were introduced in detail, and the data preprocessing method was described. On this basis, a siamese network model based on LeNet-5 was constructed, including a subnet, a comparison module, an output module and a training module. Finally, the result analysis shows that the

constructed siamese network model realizes image recognition on the MNIST, Fashion-MNIST and CIFAR-10 datasets. Comparative experiments show that the siamese networks have higher accuracy and faster recognition speed than other traditional algorithms. In addition, the siamese networks have good recognition performance on different datasets, which fully proves their strong generalization ability. Finally, the parameter sensitivity analysis was carried out, and the influence of learning rate and batch size on model performance was discussed, providing an important basis for model optimization.

Although some progress has been made in the research on image recognition based on siamese networks, this study has some limitations. Future research could focus on semi-supervised and unsupervised learning to overcome the dependence on large amounts of annotated data, enabling the siamese network to better adapt to different data scenarios. Real-time performance and efficiency optimization could also become important areas of research. This would make it easier for siamese networks to handle large amounts of data by making the algorithms better and the hardware faster. Furthermore, the research on multimodal fusion could open up new development opportunities for siamese networks. It is expected that the recognition performance and generalization ability of the siamese networks could be further improved by fusing information from different modalities.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. D. Luo, "Research on image recognition method based on deep learning," *J. Phys. Conf. Ser.*, vol. 1395, no. 1, p. 012008, 2019. https://doi.org/10.1088/1742-6596/1395/1/012008

[2] Z. Q. Zhou, B. Wang, J. Yang, and J. Lyu, "Fast target recognition based on local scale invariant features," *Opt. Tech.*, vol. 34, no. 5, pp. 742–745, 2008. https://doi.org/10.3321/j.issn:1002-1582.2008.05.017

[3] H. Y. Shi and N. Zhang, "Moving target detection method for single SAR image based on sparse representation and road assistance," *Acta Electron. Sin.*, vol. 43, no. 3, pp. 431–439, 2015. https://doi.org/10.3969/j.issn.0372-2112.2015.03.003

[4] D. Galić, Z. Stojanović, and E. Čajić, "Application of neural networks and machine learning in image recognition," *Tehn. Vjesn.*, vol. 31, no. 1, pp. 316–323, 2024. https://doi.org/10.17559/TV-20230621000751

[5] X. C. Chen, "Research on deep learning algorithms and applications based on convolutional neural networks," *Zhejiang Gongshang Univ.*, 2013. https://doi.org/10.7666/d.Y2531769

[6] M. Qin, Y. Xi, and F. Jiang, "A new improved convolutional neural network flower image recognition model," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, Xiamen, China, 2019, pp. 3110–3117. https://doi.org/10.1109/SSCI44817.2019.9003016

[7] W. Jing, "Classification and identification of garment images based on deep learning," *J. Intell. Fuzzy Syst.*, vol. 44, no. 3, pp. 4223–4232, 2023. https://doi.org/10.3233/JIFS-220109

[8] H. Q. Hu, J. Lyu, and X. L. Yin, "Research and prospect of image recognition based on convolutional neural network," *J. Phys. Conf. Ser.*, vol. 1574, no. 1, p. 012161, 2020. https://doi.org/10.1088/1742-6596/1574/1/012161

[9] J. J. Valero-Mas, A. J. Gallego, and J. R. Rico-Juan, "An overview of ensemble and feature learning in few-shot image classification using siamese networks," *Multimed. Tools Appl.*, vol. 83, no. 7, pp. 19 929–19 952, 2023. https://doi.org/10.1007/s11042-023-15607-3

[10] Y. T. He, "Lung image recognition based on siamese network," *Video Eng.*, vol. 47, no. 07, pp. 24–27, 2023. https://doi.org/10.16280/j.videoe.2023.07.005

[11] J. M. Ren, N. S. Gong, and Z. Y. Han, "An improved target tracking algorithm based on siamese convolutional neural network," *J. Chin. Comput. Syst.*, vol. 40, no. 12, pp. 2686–2690, 2019. https://doi.org/10.3969/j.issn.1000-1220.2019.12.038

[12] M. T. Pei, B. Yan, H. L. Hao, and M. Zhao, "Person-specific face spoofing detection based on a siamese network," *Pattern Recognit.*, vol. 135, p. 109148, 2023. https://doi.org/10.1016/j.patcog.2022.109148

[13] C. R. Kumar, N. Saranya, M. Priyadharshini, G. E. Derrick, and R. M. Kaleel, "Face recognition using CNN and siamese network," *Meas. Sens.*, vol. 27, p. 100800, 2023. https://doi.org/10.1016/j.measen.2023.100800

[14] Y. F. Chen, Y. Wu, and W. Zhang, "Survey of target tracking algorithms based on siamese network structure," *Comput. Eng. Appl.*, vol. 56, no. 6, pp. 10–18, 2020. https://doi.org/10.3778/j.issn.1002-8331.1911-0127

[15] S. Postalcıoğlu, "Performance analysis of different optimizers for deep learning-based image recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 2, p. 2051003, 2020. https://doi.org/10.1142/S0218001420510039

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. https://doi.org/10.1109/5.726791

[17] W. S. Lu, "Handwritten digits recognition using PCA of histogram of oriented gradient," in *2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, Victoria, BC, Canada, 2017, pp. 1–5. https://doi.org/10.1109/PACRIM.2017.8121906

[18] R. Ebrahimzadeh and M. Jampour, "Efficient handwritten digit recognition based on histogram of oriented gradients and SVM," *Int. J. Comput. Appl.*, vol. 104, no. 9, pp. 10–13, 2014.

[19] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, 2018. https://doi.org/10.1016/j.neunet.2017.12.005

[20] V. T. Nguyen, "Research on neural network activation functions for handwritten character and image recognition," Ph.D. dissertation, Xidian University, 2020.

[21] D. S. Maitra, U. Bhattacharya, and S. K. Parui, "CNN based common approach to handwritten character recognition of multiple scripts," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015, pp. 1021–1025. https://doi.org/10.1109/ICDAR.2015.7333916