Journal of Computers Vol. 34 No. 3, June 2023, pp. 137-150
doi: 10.53106/199115992023063403010

Ensemble Learning Network for Handwritten Digit Recognition Based on Fusion Optimized CNN

Li Cui[1], Ting-Xuan Chen[1], Ying-Qing Xia[1*], Xia Cao[1], Ling Wu[2]

[1] College of Physical Science and Technology, Central China Normal University, Wuhan 430079, China
542712822@qq.com, 2806519646@qq.com, yingqingxia_ccnu@126.com, 117227898@qq.com

[2] School of Physics and Electronic Information Engineering, Hubei Engineering University, Xiaogan 432100, China
250416220@qq.com

Received 18 May 2022; Revised 31 October 2022; Accepted 12 December 2022

Abstract. Handwritten digit recognition is an active research field. These recognition systems are faced with many challenges, including accuracy, speed and automatic extraction of complex handwriting features. In this paper, a Stacking ensemble learning model based on fusion optimized CNN is proposed, which can be effectively used for handwritten digit recognition. To better extract the features of complex handwritten digital images and maximize the reliability of the model, the Bagging strategy combined with six CNNs is used for feature extraction for the first time, and SVM is used for classification. This not only improves the accuracy and stability of the model, but also effectively avoids over-fitting. In addition, a fusion optimization algorithm based on Adam and SGD is proposed to solve the problem that CNN falls into local optimum due to a large number of iterations. During the process of training, ASCNN can not only speed up the convergence rate in the early stage, but also reduce the oscillation phenomenon in the late stage. Extensive experimental results on the well-known MNIST and USPS handwriting image datasets demonstrate the effectiveness of the proposed model.

Keywords: ensemble learning, fusion optimization, Bagging strategy, CNN, SVM

1 Introduction

Handwritten digit recognition is an important branch of pattern recognition. It mainly studies how to automatically recognize Arabic numerals in different scenarios with the help of computers. With the rapid development of handwritten digitization, handwritten digit recognition has been applied in many practical fields. It can be used in sorting the postal cards with multilingual zip codes [1], identifying financial bills [2], recognizing handwritten digits in historical document images [3], and marking many answers in the form of online handwritten mathematical expressions [4]. With the increasing demand of recognition accuracy and speed in society and industry, it is very important to design a highly reliable and fast handwritten digit recognition system. Therefore, handwritten digit recognition is still a challenging problem in the field of computer vision and pattern recognition.

Traditional handwritten digit recognition system includes two stages: feature extraction and classification. Most of them use shallow structures to deal with computing units and limited kinds of samples, such as support vector machines (SVM) [5]. Faced with complex classification problems, the generalization ability and performance of SVM is insufficient when the samples have rich meanings. Recently, Convolutional Neural Network (CNN) [6-9] has been widely used in the field of computer vision because of its excellent performance, and has made outstanding achievements in the accuracy of various machine learning tasks. Researchers try to solve the problem of handwritten digit recognition by combining CNN and SVM for better results. [10] uses CNN's acceptance domain as an automatic feature extractor and SVM as a classifier to form a combined model, which has a recognition accuracy of 99.28% on MNIST. A CNN-SVM model with dropout technique for offline Arabic handwritten recognition (OAHR) was proposed in [11], and the dropout rate of the CNN-SVM was 5.83% on the HACDB dataset. In [12], features were retrieved automatically based on CNN, and unknown patterns were identified by SVM. The high reliability of the proposed system on MNIST has been achieved through rejection rules, with recognition rate of 99.81% without rejection and 94.40% with rejection. From the discussion of the above literature, we find that most of the ensemble models of handwritten digit recognition are based on the two-layer

* Corresponding Author

structure of CNN-SVM. The first layer is used for feature extraction, and the second layer for classification. This two-layer structure not only improves the classification performance of the model, but also avoids the reconstruction of complex feature extraction and data processing in traditional recognition methods. Although CNN-SVM can automatically extract complex handwritten digit image features, there are still some potential challenges. (1) In these handwritten digit recognition networks of CNN-SVM, simply combining two weak learners may lead to over-fitting of the model and its unstable performance. (2) Poor performance of a single base classifier may reduce the effectiveness of the whole ensemble learning model.

To solve the problem that CNN-SVM may make the performance of the model unstable and produce over-fitting, some scholars have proposed many new algorithms, such as ensemble learning algorithm based on Bagging strategy [13]. Researchers found that it can be combined with other classification algorithms to improve the accuracy and stability of the model. At the same time, it can effectively avoid over-fitting by reducing the variance of the base classifier [13]. The strategy of Bagging has been used by many deep ensemble learning networks to deal with various technical problems such as fuel cell optimization [14], load detection [15], water quality detection [16] and medicine [17-18]. In [19], Imran Ul Haq et al. propose a new deep convolution neural network based on feature fusion and ensemble learning strategy to improve the abnormality detection in mammography. It uses fusion technology to extract features, and uses three different classifiers for ensemble learning. [20] developed another evolutionary bagged ensemble learning framework that utilizes evolutionary algorithms to randomly arrange and update data in bags in order to iteratively enhance the diversity of fusions. In addition, Prasanth Sikakollu et al. [21] developed a ensemble network composed of four CNNs with super-pixel smoothing for the task of HSI classification, which uses the diversity among classifiers to derive the optimal number of features. These experimental results show that the ensemble learning algorithm based on Bagging strategy is helpful to obtain excellent results in computer vision. However, there is little discussion about the application of ensemble learning model based on Bagging strategy in handwritten digit recognition. We hope to further analyze and explore the framework of ensemble learning by introducing Bagging strategy so as to improve the reliability and stability of handwritten digit recognition system.

In addition, the model of ensemble learning can be made to perform better by improving the performance of a single base classifier. In general, the second layer of the ensemble model does not use the original data for training, but only depends on the training results of the first layer. Therefore, the performance of CNN as the first layer of base classifier is particularly important. In this paper, we pay more attention to the performance improvement of CNN. Each deep learning network has its own setup process, which is used to learn from data and improve performance. These settings are related to the hyper-parameters in the deep learning model, which have great influence on the training time, calculation cost and performance of the model [22]. It is a problem worthy of attention to choose an appropriate optimization algorithm to optimize the hyper-parameters and make CNN get the optimal solution quickly. Recently, researchers have made some attempts to improve the performance of CNN. An ensemble algorithm based on stochastic gradient descent (SGD) with hot restart mechanism is proposed [23]. This algorithm uses the fusion method and SGD to obtain different groups of classifiers needed for integration in a single training process, which takes the same or less training time compared with that of a single CNN. Xin Yin et al [24] put forward an ensemble CNN-Adam-BO model to realize real-time prediction of rock burst intensity. In the process of modeling, adaptive moment estimation and Bayesian are used to optimize hyper-parameters of CNN and discard probability respectively. They show that the improved optimization algorithm has better performance than other existing technologies in terms of classification accuracy and computing resource usage. However, as we all know, Adam optimization algorithm has two main problems. On the one hand, it leads to the oscillation of the learning rate in the later stage of training, which leads to the inability of the model to converge. On the other hand, it may fall into the local optimal solution and miss the global optimal solution [25]. SGD is a traditional gradient descent optimization algorithm. Although the final convergence result of SGD is usually better, it has a slow convergence speed. Therefore, combining the advantages of the two algorithms, we propose a new Adam-SGD fusion optimization algorithm of CNN (ASCNN). ASCNN accelerates the convergence speed of CNN in the early training and reduces the oscillation phenomenon in the later training.

To sum up, this paper proposes an ensemble learning model based on fusion optimization of CNN, which is called ASCNNs-SVM for short. It connects two layers by Stacking ensemble method. The first layer combines six ASCNNs with different structures based on Bagging strategy to obtain features. The second layer uses SVM as classifier. In addition, to improve the performance of the base classifier, this paper proposes an Adam-SGD fusion optimization algorithm to train a single CNN. Our contributions in this paper are as follows:

1. ASCNN combines the advantages of Adam and SGD algorithms to speed up the convergence rate of CNN in the early training stage and reduce the oscillation phenomenon in the later training stage. ASCNN is superior to Adam and SGD in speed, stability and accuracy.
2. ASCNNs-SVM combines six weak classifiers of ASCNN into one strong classifier by Bagging strategy, which reduces the variance of the base classifier and avoids over-fitting. At the same time, the generalization error of the model, as well as the accuracy, is improved.
3. ASCNNs-SVM improves the accuracy of the model through an ensemble method of Stacking. It is superior to single CNN and SVM in handwritten digit recognition performance.

The structure of the rest of this paper is as follows. The related work of CNN and ensemble learning are briefly reviewed in Section 2. And the ASCNNs-SVM algorithm will be presented in Section 3. In Section 4, we report the experimental results of ASCNN and ASCNNs-SVM. Finally, summarization is given in section 5.

2 Related Work

In this section, LeNet-5 [26] was first introduced successfully, and then the ensemble learning method and CNN-SVM are briefly reviewed.

2.1 LeNet-5

LeNet-5 [26] is a well-known and very effective CNN model for handwriting recognition, which was proposed by Yan Le Cun in 1998. The structure of LeNet-5 is shown in Fig. 1. The purpose of the convolution layer is to extract different features of input images. The first convolution layer (C1) may only extract some low-level features, and more convolution layers (C3 and C4) can iteratively obtain more complex features from low-level ones. The pooling layers (S2 and S4) are actually down-sampling, which can convert a higher-resolution picture into a lower-resolution picture. At the same time, the pooling layer can further reduce the number of nodes in the fully connected layer, thereby achieving compression of data and reducing the parameters of the whole neural network. The fully connected layer (F6) transforms the two-dimensional feature map output by convolution into one-dimensional vector, which can be highly purified and easily handed over to the final classifier for classification. Compared with the traditional neural network, LeNet-5 can make good use of the structural information about the image to greatly reduce the complexity and parameters of the network due to the local connection and shared weight of the convolution layer [10-12].

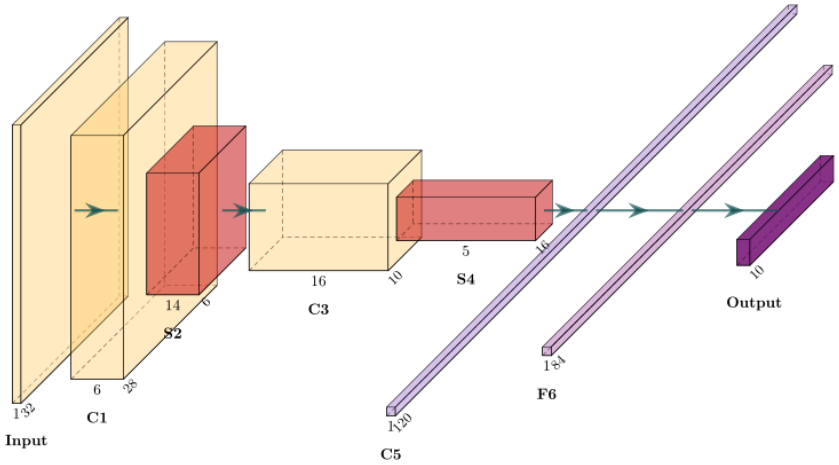


Fig. 1. LeNet-5

2.2 Ensemble Learning

In machine learning, it is difficult to find a universal learner to solve all learning problems. A suitable learner may be obtained with sufficient prior knowledge. Ensemble learning [27] can greatly improve the generalization ability of a single classifier by combining multiple base classifiers. Therefore, ensemble learning has become a hot spot of the machine field. Ensemble learning method can be divided into Boosting [28] and Bagging [13] according to the strength of the dependency between the base classifiers. The Stacking [13] algorithm proposed by Wolpert is a typical representative of ensemble learning strategies. The primary idea of Stacking method is to train the main classifier from the original dataset to obtain the prediction set, which is used for the learning of the secondary classifier. The above operation can be repeated for multiple stacking, which means that the two integration methods mentioned above will be integrated by using a suitable combination strategy in theory. Therefore, the Stacking algorithm adopts the idea of multi-stage learning, which is a more complex combination strategy.

2.3 CNN-SVM

The two main operations of machine learning technology are feature extraction and classification. In CNN architecture, feature extraction and classification techniques are combined into one model, thus eliminating the need for a separate method of feature extraction [29]. SVM [30] is an advanced classification algorithm based on statistical learning theory and structural risk minimization principle through manual feature extraction, which has good generalization performance. CNN-SVM replaces the last output layer of the CNN model with the SVM classifier, which enables automatic feature extraction and achieves high performance. This hybrid model combines the advantages of two classifiers to compensate for the limitations of CNN and SVM, and has been used in handwritten digit recognition. The CNN-SVM [31] model structure is shown in Fig. 2.

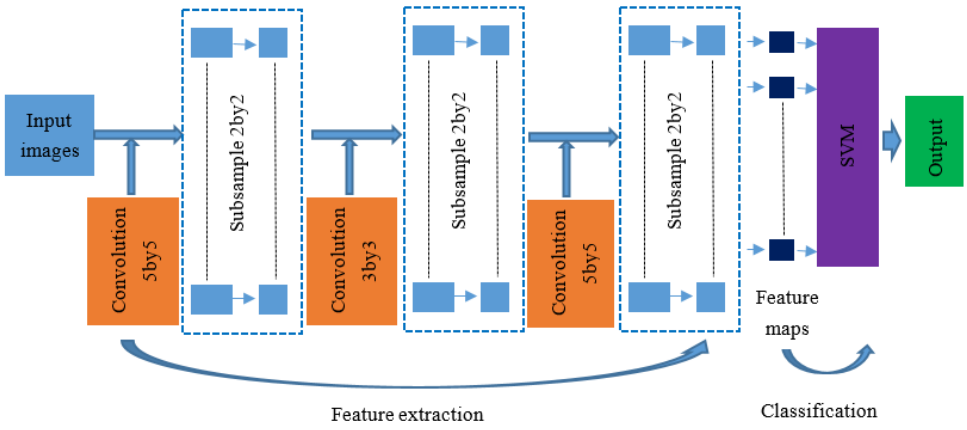


Fig. 2. Architecture of CNN-SVM

3 ASCNNs-SVM

In this section, the details of ASCNNs-SVM are explained. First, the model of ASCNN is described. Then, the ASCNNs-SVM is explained from the perspective of ensemble learning.

3.1 ASCNN

a) Structure design of CNN

In general, optimizing the preprocessing method and selecting the appropriate complex structure of CNN can improve the recognition performance of the network. However, these attempts are often accompanied by more training time and higher hardware performance requirements. Moreover, not all complex structures of CNN are better for datasets, because more complex structure is more likely to lead to over-fitting problems in training. From another point of view, the ensemble model improves the recognition performance by combining several CNNs with simple structure. After experimental exploration, the final structure of CNN is shown in Table 1. It shows that CNN includes three convolution layers, three sampling layers and two fully connected layers. The feature map input in the experiment has a small size (28×28). The feature map becomes smaller after convolution and pooling. This process may result in the loss of original feature mapping information and is not conducive to network training. Therefore, CNN uses edge filling technology in the convolution layer to ensure that the deep input feature map can retain sufficient edge information.

Table 1. Structure of CNN

Layer number	Layer type	Output shape	Convolution kernel	Step	Dropout
0	Input layer	28×28×1	—	—	—
1	Convolutional layer	28×28×32	5×5	1	—
2	Sampling layer	14×14×32	2×2	2	—
3	Convolutional layer	14×14×64	3×3	1	—
4	Sampling layer	7×7×64	2×2	2	—
5	Convolutional layer	7×7×128	5×5	1	—
6	Sampling layer	3×3×128	2×2	2	—
7	Fully-connected layer	1024×1	—	—	30%
8	Fully-connected layer	256×1	—	—	30%
9	Output layer	10×1	—	—	—

b) ASCNN

The optimization algorithm is called an optimizer, which is used to minimize the loss function of a training process of CNN. SGD maintains a constant learning rate in training to update all weights. Adam designs independent adaptive learning rates for different parameters by calculating First-Order and Second-Order Moment Estimation of the gradient. ASCNN combines the advantages of Adam and SGD optimizers so that it can optimize CNN. The combinatorial optimization algorithm uses the early stopping method to switch the time for the two optimizers. In this process, the Adam optimizer is responsible for fast convergence, and the SGD optimizer is responsible for finding the optimal solution. The training process of ASCNN is shown in Table 2.

Table 2. ASCNN model training process

Algorithm 1. Model training

INPUT: A set of training samples images, P is set as the maximum eopch tolerance of the Early-Stopping method.
Output: A set of optimal decision-making behavior for dataset.
Data set is divided into training set, verification set and test set.
While P <= Maximum,
Adam was used to optimize the CNN of the training set.
  The value of loss function of the CNN was calculated on the verification set.
  If the value of loss function < Minimum,
    Minimum = the value of loss function.
End
The SGD is used for CNN.
Obtain the final model of combined Adam-SGD optimization.

3.2 ASCNNs-SVM

Ensemble learning is a common method to improve the recognition performance of network. The main idea of ensemble learning is to integrate some weak classifiers with poor recognition performance into a strong classifier with good recognition performance. The biggest advantage of ensemble learning is that it can break through the bottleneck of recognition performance of the original model without designing a very complex model structure. The proposed classification model of ASCNNs-SVM is composed of two-level classifiers through the Stacking method. The first layer classifier is mainly used to automatically extract handwritten digit features and is com-

posed of six ASCNNs based on Bagging strategy. The second layer classifier is composed of SVM, which is mainly used for final classification. The six ASCNNs with different structures are named CNN1, CNN2, CNN3, CNN4, CNN5, and CNN6. CNN1 has the same structure as CNN in Table 1. On the basis of the CNN1, the remaining five network models are designed by changing the size and number of convolution kernels. Normally, the convolution layer close to the input, such as the first convolution layer, will find some common features. In the first layer of handwritten digit recognition, we set a relatively small number of convolution kernels to find common features such as “horizontal lines”, “vertical lines”, and “slashes”, which are called basic features. After max pooling, the number of convolution kernels of the second convolution layer is set to be a bigger one. This process will enable some relatively complex features to be found, such as “horizontal fold”, “left semicircle”, “right semicircle” and other features. The more the convolution kernels are used, the more detailed the characteristics of the sample will be. This will make the process of classification easier to implement. Therefore, the number of convolution kernels of different convolution layers in the network will increase as the number of layers deepens. In order to alienate the feature extraction level of the network and improve the classification effect, different combinations of convolution kernels are used in different base classifiers. The structures of five CNN models are shown in Table 3.

Table 3. The structures of five CNN

CNN1					CNN2				
Layer number	Layer type	Output shape	Filter size	Dropout	Layer number	Layer type	Output shape	Filter size	Dropout
0	Input layer	28×28×1	—	—	0	Input layer	28×28×1	—	—
1	Convolutional layer	28×28×32	5×5	—	1	Convolutional layer	28×28×64	5×5	—
2	Sampling layer	14×14×32	2×2	—	2	Sampling layer	14×14×64	2×2	—
3	Convolutional layer	14×14×64	3×3	—	3	Convolutional layer	14×14×96	5×5	—
4	Sampling layer	7×7×64	2×2	—	4	Sampling layer	7×7×96	2×2	—
5	Convolutional layer	7×7×128	5×5	—	5	Convolutional layer	7×7×128	5×5	—
6	Sampling layer	3×3×128	2×2	—	6	Sampling layer	3×3×128	2×2	—
7	Fully-connected layer	1024×1	—	30%	7	Fully-connected layer	1024×1	—	30%
8	Fully-connected layer	256×1	—	30%	8	Fully-connected layer	256×1	—	30%
9	Output layer	10×1	—	—	9	Output layer	10×1	—	—
CNN3					CNN4				
Layer number	Layer type	Output shape	Filter size	Dropout	Layer number	Layer type	Output shape	Filter size	Dropout
0	Input layer	28×28×1	—	—	0	Input layer	28×28×1	—	—
1	Convolutional layer	28×28×48	5×5	—	1	Convolutional layer	28×28×64	5×5	—
2	Sampling layer	14×14×48	2×2	—	2	Sampling layer	14×14×64	2×2	—
3	Convolutional layer	14×14×64	5×5	—	3	Convolutional layer	14×14×96	3×3	—
4	Sampling layer	7×7×64	2×2	—	4	Sampling layer	7×7×96	2×2	—
5	Convolutional layer	7×7×96	5×5	—	5	Convolutional layer	7×7×128	5×5	—
6	Sampling layer	3×3×96	2×2	—	6	Sampling layer	3×3×128	2×2	—
7	Fully-connected layer	1024×1	—	30%	7	Fully-connected layer	1024×1	—	30%
8	Fully-connected layer	256×1	—	30%	8	Fully-connected layer	256×1	—	30%
9	Output layer	10×1	—	—	9	Output layer	10×1	—	—
CNN5					CNN6				
Layer number	Layer type	Output shape	Filter size	Dropout	Layer number	Layer type	Output shape	Filter size	Dropout
0	Input layer	28×28×1	—	—	0	Input layer	28×28×1	—	—
1	Convolutional layer	28×28×32	5×5	—	1	Convolutional layer	28×28×64	5×5	—
2	Sampling layer	14×14×32	2×2	—	2	Sampling layer	14×14×64	2×2	—
3	Convolutional layer	14×14×64	3×3	—	3	Convolutional layer	14×14×96	3×3	—
4	Sampling layer	7×7×64	2×2	—	4	Sampling layer	7×7×96	2×2	—
5	Convolutional layer	7×7×128	3×3	—	5	Convolutional layer	7×7×128	3×3	—
6	Sampling layer	3×3×128	2×2	—	6	Sampling layer	3×3×128	2×2	—
7	Fully-connected layer	1024×1	—	30%	7	Fully-connected layer	1024×1	—	30%
8	Fully-connected layer	256×1	—	30%	8	Fully-connected layer	256×1	—	30%
9	Output layer	10×1	—	—	9	Output layer	10×1	—	—

Furthermore, the method of Stacking is a typical representative in ensemble learning. In the first learning stage, the six basic classifiers are integrated into one through the Bagging strategy. The training and testing data sets in the second learning stage are obtained depending on the primary classifier. Then the second stage learning is implemented by SVM. Finally, the network of ASCNNs-SVM is constituted. The structure of ASCNNs-SVM is shown in Fig. 3.

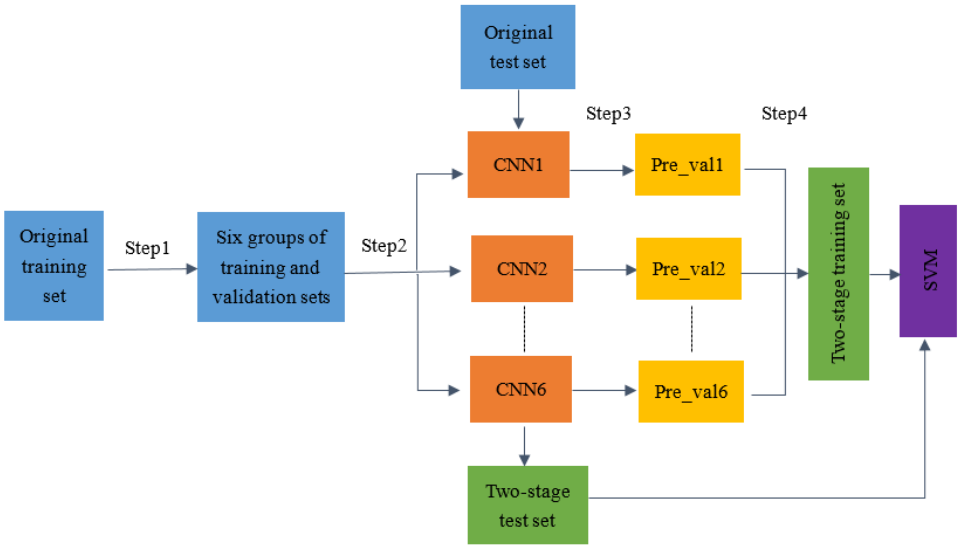


Fig. 3. The schematic diagram of ASCNNs-SVM

The specific training steps of ASCNNs-SVM based on Bagging strategy are as follows:

Step 1: The original training set is divided into six parts, five of which are used as training set and one as verification set. In this way, six groups of training and verification sets to be used in the first stage of learning can be obtained. The original test set is still used for testing in the first stage of learning.

Step 2: The six training sets in step 1 are used in the training of CNN1, and the prediction set of the corresponding verification set and testing set are saved according to each training result.

Step 3: Splice the prediction sets of the verification set obtained in step 2 to obtain pre_val1, and average and combine the prediction sets of the test set to obtain pre_test1. For each single-stage base classifier, repeat steps 2 and 3 to get pre_val1-6, and pre_test1-6.

Step 4: pre_val1-6 are spliced as the training set to train the SVM classifier for the second stage of learning, and pre_test1-6 are spliced as the test set for the second stage of learning.

4 Experimental Setup and Discussion of Results

In this section, we verify the performance of ASCNNs-SVM in handwritten digit recognition. The performance of Adam, SGD and Adam-SGD fusion optimization algorithms are compared at first. Then, the second-stage classifier of the ensemble model is replaced with the classic K-Nearest Neighbors (KNN) [32], Linear Discriminant Analysis (LDA) [33], Logistic Regression (LR) [34], Random Forest (RF) [35] and fully connected layer + Softmax [26] and experiments are conducted to analyze the effectiveness of ASCNNs-SVM. Finally, compared with ANN [26], CNN [36], CNN-SVM [10], SVM [10], SNN [37], Q-ADBN [6], CR-MSOM [7], integrated learning based on ASCNNs-SVM proved to be superior. The Keras deep learning framework is used for modeling and testing. All the experiments are implemented in PYTHON3.6 on a PC with Intel Core(TM) core i5 CPU (2.5 GHz) with 24 GB RAMS.

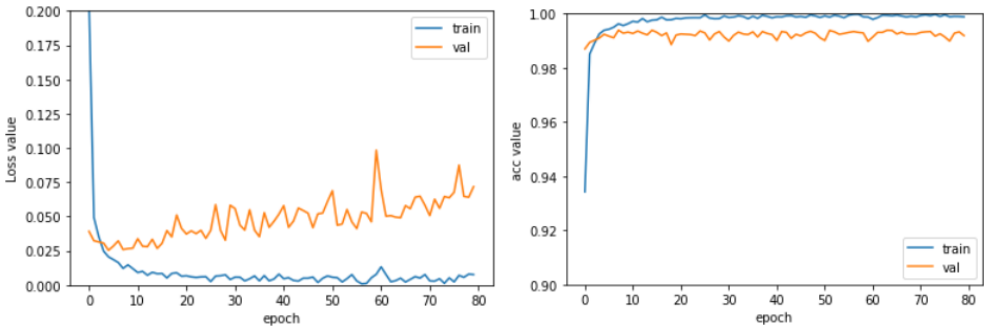
4.1 Data Sets

MNIST [38] is a recognized handwritten digit data set, containing 70,000 samples of handwritten Arabic numerals ranging from 0 to 9. Each image has been standardized to a 28×28 gray image, with the number in the center of the image. MNIST is divided into 50,000 training samples, 10,000 verification samples and 10,000 test samples.

USPS [39] is the handwritten digit recognition database of the U.S. Postal service, which contains 9298 handwritten digital images. The size of each gray image is 16×16, and its gray value has been normalized. USPS is divided into 6640 training samples, 1329 validation samples and 1329 test samples.

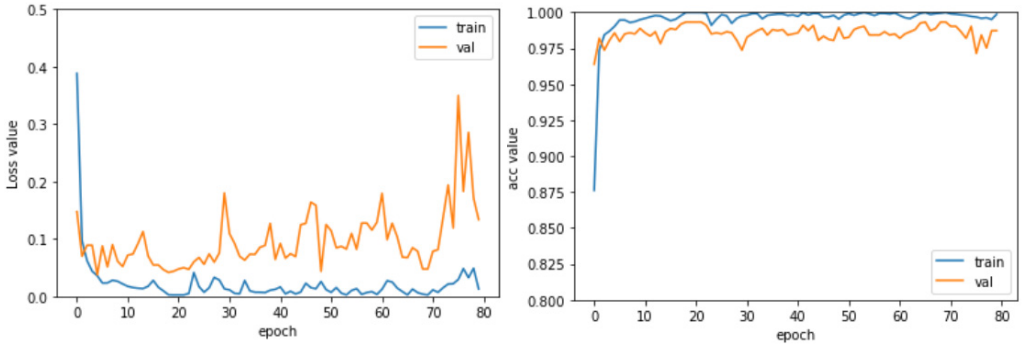
4.2 ASCNN Experiment and Result Analysis

To verify that the fusion optimization algorithm can quickly converge the network model and reduce the over-fitting phenomenon, the Adam-SGD is used to optimize CNN1. The experiments uniformly select the ReLU function as the activation function of the convolutional layer in order to reduce over-fitting. The edge filling operation is used in the convolutional layer and the pooling layer. The Dropout method is used in the fully connected layer. The value of batch_size is set to 128 in MNIST and 32 in USPS. The value of epoch is set to 80, and P is set to 5.



(a) Change of loss function in training process (b) Change of accuracy in training process

Fig. 4. Performance of Adam on MNIST



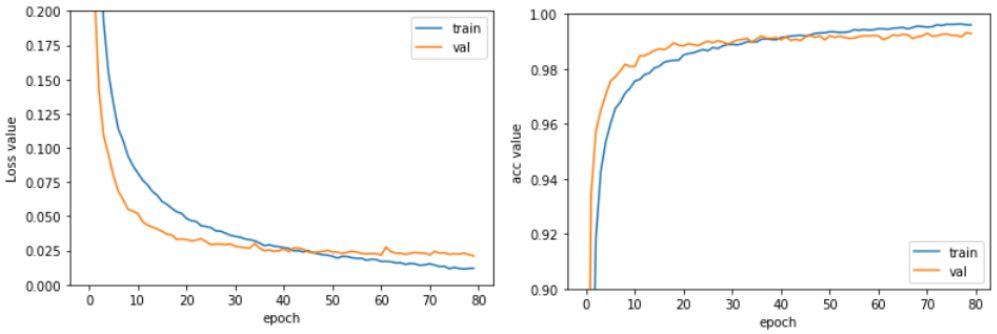
(a) Change of loss function in training process (b) Change of accuracy in training process

Fig. 5. Performance of Adam on USPS

As is shown in Fig. 4, the loss functions of training and validation sets drop rapidly in the early stage by using Adam algorithm optimization of network training on MINIST. After about 15 epochs, the loss function of validation set oscillates significantly and generally shows an upward trend. This phenomenon shows that there is over-fitting in the training process. The accuracy of training and verification sets reached 0.99 quickly, but the accuracy of verification set fluctuated greatly in the later stage, which made it difficult for the network to con-

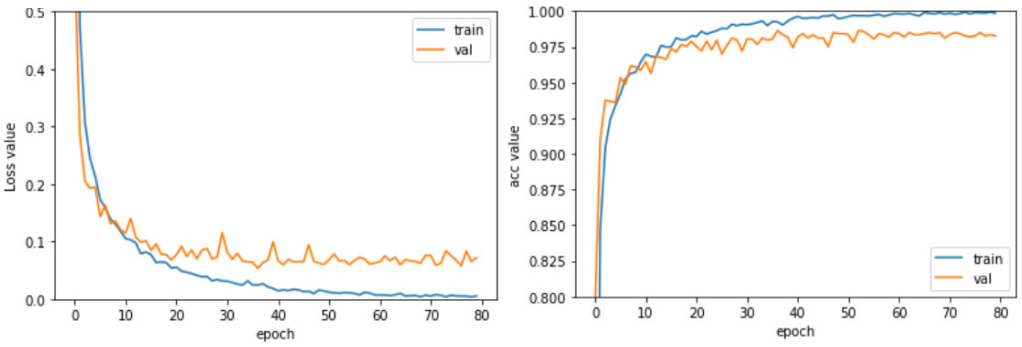
verge. It can be seen from Fig. 5 that the experimental results of USPS have similar features. In the initial stage of training, the loss functions of training and validation sets dropped rapidly and their accuracies rose rapidly to 0.97, but then the curve of the loss function of validation set oscillated violently, resulting in poor network convergence.

In the early stage of network training on MNIST and USPS, compared with Adam, the loss function of SGD decreases more slowly, indicating that Adam optimization algorithm can help the network to converge faster, which is shown in Fig. 6 to Fig. 7. However, the loss function curve obtained by using SGD has been in a downward trend without obvious earthquake phenomenon. It shows that the convergence is slow, but the training process is relatively stable in network training. The same is true for accuracy. The accuracies of training and verification sets trained by SGD are lower than that of Adam. The accuracy of MNIST reached 0.99 after 40 epochs. The accuracy of USPS reached 0.97 after 20 epochs, but with very small fluctuation and better convergence.



(a) Change of loss function in training process (b) Change of accuracy in training process

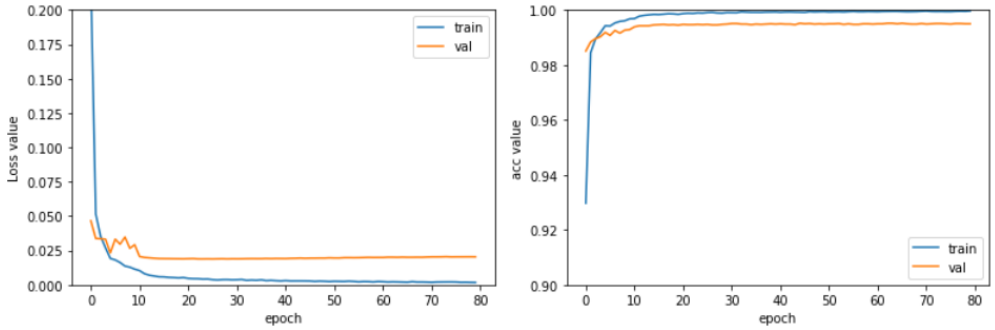
Fig. 6. Performance of SGD on MNIST



(a) Change of loss function in training process (b) Change of accuracy in training process

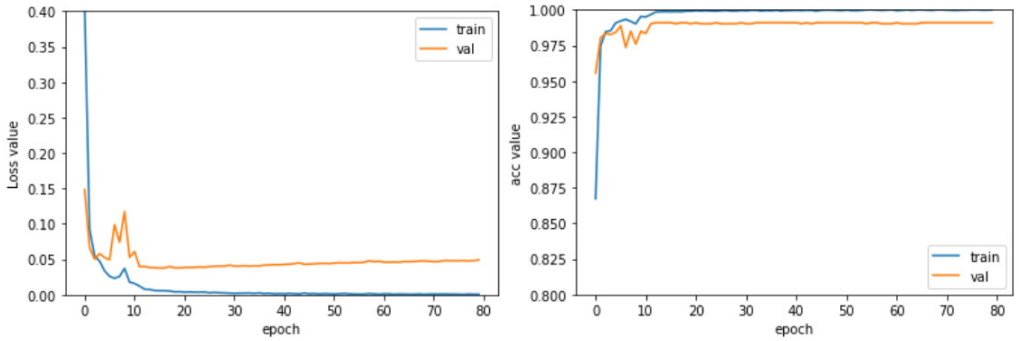
Fig. 7. Performance of SGD on USPS

Fig. 8 to Fig. 9 shows the results of CNN training based on the Adam-SGD optimization algorithm in MNIST and USPS. In the early stage of training, the loss function drops quickly and the accuracy rises quickly by using the Adam optimizer. The accuracy has reached a higher value when the loss function has a more obvious seismic oscillation and rebound trend. Then, MNIST and USPS use SGD to train the network after the 10th and 11th epoch respectively according to the judgment of the early stopping method. As can be seen from the change curves of loss function and accuracy, the network could converge quickly. On MNIST, compared with Adam algorithm, Adam-SGD began to converge after the 11th epoch. Meanwhile, the convergence time of Adam-SGD on MNIST is 29 epochs shorter than that of SGD. On USPS, study based on Adam-SGD has a similar conclusion.



(a) Change of loss function in training process (b) Change of accuracy in training process

Fig. 8. Performance of ASCNN on MNIST



(a) Change of loss function in training process (b) Change of accuracy in training process

Fig. 9. Performance of ASCNN on USPS

From the perspective of the effectiveness of the algorithm, the network recognition performance finally obtained by using the Adam-SGD fusion optimization algorithm on MNIST and USPS is superior to that obtained by using Adam or SGD alone (see Table 4). On MNIST and USPS, the accuracy of Adam-SGD is obviously superior to the other two algorithms, with the highest improvement of 0.91%. Moreover, the recognition performance obtained by using SGD on MNIST is better than that obtained by using Adam. However, the performance on USPS is the opposite, which is likely to be that SGD has fallen into a local optimal solution during network training. The experimental results show that Adam-SGD optimization algorithm improves the accuracy of CNN obviously due to Adam's fast initial training speed and SGD's contribution to the stable convergence of the model. At the same time, the training time of CNN is reduced.

Table 4. Accuracy of three optimization algorithms

Data set	Optimization algorithm	Accuracy
MNIST	Adam	99.19%
	SGD	99.29%
	Adam-SGD	**99.45%**
USPS	Adam	98.27%
	SGD	97.59%
	Adam-SGD	98.50%

In summary, the Adam-SGD fusion optimization algorithm can help the network converge to a good result faster, which is better than the Adam or the SGD optimization algorithm in terms of speed, stability and accuracy.

4.3 Analysis of the Results of ASCNNs-SVM and Classic Classifiers as the Second-stage Classifier Network

ASCNNs-SVM conducts experiments by integrating six CNNs of Adam-SGD fusion optimized and SVM to construct a network. Compared with the classic classifier as the second-stage classifier, the superiority of ASCNNs-SVM is verified. First, MNIST is divided into 60,000 training samples and 10,000 test samples. The training samples are divided into 6 groups on average, which form 6 groups of training set of 50,000 samples and validation set of 10,000 samples respectively. Similarly, USPS is divided into 7, 968 training samples and 1330 test samples, forming 6 groups of training set of 6640 samples and verification set of 1328 samples. Then, the six base classifiers are trained separately with six groups of training and verification sets. The experiment parameters are the same as those in 4.2. Table 5 to Table 6 shows the classification accuracy of each base classifier of under different data grouping conditions.

Table 5. The accuracy of base classifiers in different data groups on MNIST

Base classifier	First group	Second group	Third group	Fourth group	Fifth group	Sixth group	Average accuracy
CNN1	99.33%	99.36%	99.36%	99.37%	99.36%	99.44%	99.37%
CNN2	99.49%	99.40%	99.39%	99.32%	99.31%	99.47%	99.40%
CNN3	99.39%	99.34%	99.39%	99.39%	99.41%	99.45%	99.40%
CNN4	99.39%	99.37%	99.31%	99.46%	99.36%	99.42%	99.39%
CNN5	99.48%	99.34%	99.34%	99.32%	99.43%	99.35%	99.38%
CNN6	99.39%	99.38%	99.48%	99.41%	99.42%	99.47%	99.43%
Average accuracy	99.41%	99.37%	99.38%	99.38%	99.38%	99.43%	99.39%

Table 6. The accuracy of base classifiers in different data groups on USPS

Base classifier	First group	Second group	Third group	Fourth group	Fifth group	Sixth group	Average accuracy
CNN1	98.57%	98.19%	98.57%	98.19%	97.74%	98.12%	98.23%
CNN2	98.57%	98.12%	97.89%	98.42%	97.97%	98.34%	98.22%
CNN3	98.57%	98.27%	98.72%	98.04%	98.34%	98.04%	98.33%
CNN4	98.12%	98.12%	97.82%	98.42%	98.34%	98.41%	98.21%
CNN5	98.65%	98.27%	98.04%	98.57%	98.27%	98.34%	98.36%
CNN6	98.42%	98.19%	98.72%	98.79%	98.19%	98.34%	98.44%
Average accuracy	98.48%	98.19%	98.29%	98.41%	98.14%	98.27%	98.30%

It can be seen from Table 5 to Table 6 that different data sets and network structures will have an impact on the experimental results. The training set and test set of the second stage of stacking are generated under the influence of these two factors. The average prediction accuracy of 36 models on MNIST was 99.39%, and it's 98.30% on USPS. For the same base classifier, the training data of each group is not exactly the same, so the performance on the test set is different after different groups of data are trained. However, every sample in the original training data set participated in all the grouping training for a single base classifier. Therefore, the Bagging strategy is used to train the base classifier, which not only makes full use of the original training data set, but also reduces the dependence of the model on the selection of verification set. This method fully guarantees the stability of the integrated model and avoids over-fitting.

In that next experiment, we choose different classifiers as the learners in the second stage of the Stacking ensemble learning model. The six classifiers are KNN, LR, LDA, SVM, RF and fully connected layer+softmax. The accuracy of the test set is shown in Table 7 to Table 8.

Table 7. Network accuracy of different two-stage classifiers on MNIST

References of two-stage classifier	Two-stage classifier	Accuracy
[32]	KNN	99.56%
[34]	LR	99.55%
[33]	LDA	99.54%
[35]	RF	99.55%
[26]	Full Connectivity Layer +Softmax	99.57%
This paper	SVM	99.59%

Table 8. Network accuracy of different two-stage classifiers on USPS

References	Two-stage classifier	Accuracy
[32]	KNN	98.81%
[34]	LR	98.95%
[33]	LDA	98.81%
[35]	RF	98.87%
[26]	Full Connectivity Layer +Softmax	98.93%
This paper	SVM	**99.02%**

As is shown in Table 5 to Table 8, the recognition performance of the ensemble model based on stacking method is superior to that of the single base classifier. On MNIST, the ensemble model based on Stacking method can reach 99.59%, while the base classifier can reach 99.39%. On USPS, this model has a similar result. In addition, it is easy to see from Table 7 to Table 8 that the classification accuracy of MNIST and USPS with SVM as the second stage classifier is the highest, with 99.59% and 99.02% respectively. Facts have proved that the ensemble learning model of ASCNNs-SVM is obviously superior to the other five methods in handwritten digit recognition. Therefore, our proposed method not only improves the accuracy and stability of the ensemble model, but also avoids over-fitting.

4.4 Comparison and Discussion with the Latest Methods

This experiment aims to understand and analyze the effectiveness and generalization of the proposed ASCNNs-SVM method in handwritten digital image recognition. Compared with the latest handwritten digit recognition methods [6-7, 10, 26, 36-37] on MNIST, the accuracy is shown in Table 9. We noticed that ASCNNs-SVM achieved the highest recognition accuracy of 99.59%; while CNN and SVM achieved the lowest recognition results, 96.53% and 97.68% respectively. In addition, since Q-ADBN uses an adaptive deep auto-encoder to extract the features of the original image and regards the extracted features as the current state of the Q-learning algorithm, Q-ADBN obtains the second highest recognition rate of 99.18%. ANN (LeNet-5), CR-MSOM, SNN and other methods are also applicable to handwriting classification. Among them, the performance of ANN (LeNet-5) is slightly better than other methods, with a recognition rate of 99.05%. These results show that the accuracy of ASCNNs-SVM is obviously better than that of other classifiers. Therefore, ASCNNs-SVM ensemble learning model has better accuracy and generalization than single recognition network [6-7, 10, 26, 36-37] in handwritten digit recognition. In addition, compared with the simple ensemble model of CNN-SVM, the accuracy of ASCNNs-SVM is improved by 0.71%, for ASCNNs-SVM is affected by Bagging strategy and Adam-SGD optimization algorithm. In summary, ASCNNs-SVM network can provide the best recognition accuracy and better performance in handwritten digit recognition.

Table 9. Accuracy of the latest method on MNIST

References	Method	Accuracy
1998 [26]	ANN (LeNet-5)	99.05%
2014 [36]	CNN	96.53%
2014 [7]	CR-MSOM	99.03%
2018 [37]	SNN	98.17%
2018 [6]	Q-ADBN	99.18%
2020 [10]	SVM	97.68%
2020 [10]	CNN-SVM	98.88%
This paper	ASCNNs-SVM	**99.59%**

5 Conclusion

Handwritten digit recognition has been widely used in finance, education, logistics and other fields. Therefore, this paper studies handwritten digit recognition based on CNN. To improve the training speed and recognition accuracy of CNN, Adam-SGD fusion optimization algorithm is proposed. The experimental results prove that this method can make the network converge to a better result faster on MNIST and USPS. To further improve

the recognition performance, an ensemble learning model is constructed on the basis of ASCNN. In this paper, six different ASCNNs are designed as the base classifiers of ensemble learning, which are integrated by Bagging strategy. The final ensemble learning model is composed of six ASCNNs and a single SVM based on stacking method. Compared with the recent models of handwritten digit recognition, ASCNNs-SVM achieves better recognition accuracy on MNIST and USPS. A large number of experimental results show that the performance of ASCNNs -SVM is superior.

With the rapid development of handwriting digitalization, handwritten digit recognition has been applied in many fields. In order to improve the performance of handwritten digit recognition, more work needs doing. On the one hand, the six base classifiers designed in this paper are all CNNs. To increase the difference of base classifiers, different kinds of classifiers will be used as base classifiers for ensemble learning. This paper will try to quantify the difference of the base classifier, and further study the influence of the difference degree of the base classifier on the performance of the ensemble model. On the other hand, the data set used in this experiment is a common set. To improve the performance of handwritten digit recognition in some specific application scenarios, we can collect handwritten numeral samples with more practical background for experiments. This prediction model will be more suitable for people with specific writing habits, such as doctors, students, teachers etc. Therefore, the future work will combine multiple aspects to improve the performance of handwritten digit recognition system.

6 Acknowledgement

Li Cui and Ting-Xuan Chen contributed equally to this work. This work is financially supported by self-determined research funds of CCNU from the colleges' basic research and operation of MOE (Grant No. CCNU19ZN020).

References

[1] D. Gupta, S. Bag, CNN-based multilingual handwritten numeral recognition: a fusion-free approach, Expert Systems with Applications 165(2021) 113784.
[2] H. Gao, D. Ergu, Y. Cai, F. Liu, B. Ma, A robust cross-ethnic digital handwriting recognition method based on deep learning, Procedia Computer Science 199(2022) 749-756.
[3] H. Kusetogullari, A. Yavariabdi, J. Hall, N. Lavesson, DIGITNET: a deep handwritten digit detection and recognition method using a new historical handwritten digit dataset, Big Data Research 23(2021) 100182.
[4] H.Q. Ung, C.T. Nguyen, K.M. Phan, V.T.M. Khuong, M. Nakagawa, Clustering online handwritten mathematical expressions, Pattern Recognition Letters 146(2021) 267-275.
[5] H. Merabti, B. Farou, H. Seridi, A segmentation-recognition approach with a fuzzy-artificial immune system for unconstrained handwritten connected digits, Informatica 42(1)(2018) 95-106.
[6] J.-F. Qiao, G.-M. Wang, W.-J. Li, M. Chen, An adaptive deep Q-learning strategy for handwritten digit recognition, Neural Networks 107(2018) 61-71.
[7] E. Mohebi, A. Bagirov, A convolutional recursive modified Self Organizing Map for handwritten digits recognition, Neural Networks 60(2014) 104-118.
[8] A. Trivedi, S. Srivastava, A. Mishra, A. Shukla, R. Tiwari, Hybrid evolutionary approach for Devanagari handwritten numeral recognition using Convolutional Neural Network, Procedia Computer Science 125(2018) 525-532.
[9] D.T. Mane, U.V. Kulkarni, Visualizing and Understanding Customized Convolutional Neural Network for Recognition of Handwritten Marathi Numerals, Procedia Computer Science 132(2018) 1123-1137.
[10] S. Ahlawat, A. Choudhary, Hybrid CNN-SVM Classifier for Handwritten Digit Recognition, Procedia Computer Science 167(2020) 2554-2560.
[11] M. Elleuch, R. Maalej, M. Kherallah, A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition, Procedia Computer Science 80(2016) 1712-1723.
[12] X.-X. Niu, C.Y. Suen, A novel hybrid CNN–SVM classifier for recognizing handwritten digits, Pattern Recognition 45(4)(2012) 1318-1325.
[13] M.A. Ganaie, M. Hu, A.K. Malik, M. Tanveer, P.N. Suganthan, Ensemble deep learning: A review. <https://arxiv.org/abs/2104.02395>, 2021 (accessed 03.06.21).
[14] W. Fan, B. Xu, H. Li, G. Lu, Z. Liu, A novel surrogate model for channel geometry optimization of PEM fuel cell based on Bagging-SVM Ensemble Regression, International Journal of Hydrogen Energy 47(33)(2022) 14971-14982.
[15] X. Hu, Y. Zeng, C. Qin, D.-Z. Meng, Bagging-based neural network ensemble for load identification with parameter sensitivity considered, Energy Reports 8(supplement 13)(2022) 199-205.

[16] A. Aldrees, H.H. Awan, M.F. Javed, A.M. Mohamed, Prediction of water quality indexes with ensemble learners: Bagging and Boosting, Process Safety and Environmental Protection 168(2022) 344-361.
[17] E. Lin, C.-H. Lin, H.Y. Lane, A bagging ensemble machine learning framework to predict overall cognitive function of schizophrenia patients with cognitive domains and tests, Asian Journal of Psychiatry 69(2022) 103008.
[18] J. Lin, H. Chen, S. Li, Y.-S. Liu, X. Li, B. Yu, Accurate prediction of potential druggable proteins based on genetic algorithm and Bagging-SVM ensemble classifier, Artificial intelligence in medicine 98(2019) 35-47.
[19] I.U. Haq, H. Ali, H.-Y. Wang, C. Lei, H. Ali, Feature fusion and Ensemble learning-based CNN model for mammographic image classification, Journal of King Saud University-Computer and Information Sciences 34(6)(2022) 3310-3318.
[20] G. Ngo, R. Beard, R. Chandra, Evolutionary bagging for ensemble learning, Neurocomputing 510(2022) 1-14.
[21] P. Sikakollu, R. Dash, Ensemble of multiple CNN classifiers for HSI classification with Superpixel Smoothing, Computers & Geosciences 154(2021) 104806.
[22] S. Raziani, M. Azimbagirad, Deep CNN Hyperparameter Optimization Algorithms for Sensor-based Human Activity Recognition, Neuroscience Informatics 2(3)(2022) 100078.
[23] G. Vrbančič, V. Podgorelec, Efficient ensemble for image-based identification of Pneumonia utilizing deep CNN and SGD with warm restarts, Expert Systems with Applications 187(2022) 115834.
[24] X. Yin, Q. Liu, X. Huang, Y. Pan, Real-time prediction of rockburst intensity using an integrated CNN-Adam-BO algorithm based on microseismic data and its engineering application, Tunnelling and Underground Space Technology 117(2021) 104133.
[25] Q. Tong, G. Liang, J. Bi, Calibrating the adaptive learning rate to improve convergence of ADAM, Neurocomputing 481(2022) 333-356.
[26] L. Yann, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11) (1998) 2278-2324.
[27] X.-B. Dong, Z.-W. Yu, W.-M. Cao, Y.-F. Shi, Q.-L. Ma, A survey on ensemble learning, Frontiers of Computer Science 14(2)(2020) 241-258.
[28] M.D. Guillen, J. Aparicio, M. Esteve, Gradient Tree Boosting and the estimation of production frontiers, Expert Systems with Applications 214(2023) 119134.
[29] S. Kiranyaz, T. Ince, M. Gabbouj, Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks, IEEE Transactions on Biomedical Engineering 63(3)(2016) 664-675.
[30] C. Cortes, V. Vapnik, Support-Vector Network, Machine Learning 20(3)(1995) 273-297.
[31] T.-X. Chen, W.-M. Chen, Z.-Y. Wang, Research on digital recognition based on fusion optimization algorithm and ensemble learning, Information & Communications 4(2020) 15-17.
[32] A. Kataria, M. Singh, A review of data classification using k-nearest neighbour algorithm, International Journal of Emerging Technology and Advanced Engineering 3(6)(2013) 354-360.
[33] K. Fukunaga, Introduction to statistical pattern recognition, Academic Press, LonDon,1990 (Chapter 1).
[34] S. Menard, Logistic regression, American Statistician 58(4)(2004) 364.
[35] L. Breiman, Random Forests, Machine Learning 45(1)(2001) 5-32.
[36] B. Graham, Fractional Max-pooling. <https://arxiv.org/abs/1412.6071>, 2014 (accessed 06.08.21).
[37] S.R. Kulkarni, B. Rajendran, Spiking neural networks for handwritten digit recognition-Supervised learning and network optimization, Neural Networks 103(2018) 118-127.
[38] Y.-L. Cun, The MNIST Database of Handwritten Digits. <http://yann.lecun.com/exdb/mnist/>, 1998 (accessed 06.03.20).
[39] J.J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis & Machine Intelligence 16(5)(1994) 550-554.