Jianqiang Li[1], Weimin Shi[1], Donghe Yang[2]*

Clothing Image Classification with a Dragonfly Algorithm Optimised Online Sequential Extreme Learning Machine

DOI: 10.5604/01.3001.0014.7793

[1] **Zhejiang Sci-Tech University, Key Laboratory of Modern Textile Machinery & Technology of Zhejiang Province,**
Hangzhou, Zhejiang 310018, China

[2] **Zhejiang Sci-Tech University, School of Information Science and Technology,**
Hangzhou 310018, China,
* e-mail: yangdonghe2000@163.com

Abstract
This study proposes a solution for the issue of the low classification accuracy of clothing images. Using Fashion-MNIST as the clothing image dataset, we propose a clothing image classification technology based on an online sequential extreme learning machine (OSELM) optimised by the dragonfly algorithm (DA). First, we transform the Fashion-MNIST dataset into a data set that we extract from the corresponding grey image. Then, considering that the input weight and hidden layer bias of an OSELM are generated randomly, a DA is proposed to optimise the input weight and hidden layer bias of the OSELM to reduce the influence of random generation on the classification results. Finally, the optimised OSELM is applied to the clothing image classification. Compared to the other seven types of classification algorithms, the proposed clothing image classification model with the DA-optimised OSELM reached 93.98% accuracy when it contained 350 hidden nodes. Its performance was superior to other algorithms that were configured with the same number of hidden nodes. From a stability analysis of the box-plot, it was found that there were no outliers exhibited by the DA-OSELM model, whereas some other models had outliers or had lower stability compared to the model proposed, thereby validating the efficacy of the solution proposed.

Key words: *Dragonfly algorithm, Online Sequential Extreme Learning Machine, clothing image classification, optimised parameter.*

Introduction

Intelligent clothing image classification technology can be applied to business websites to realizes the automatic classification and retrieval of clothing, thereby saving the time and manpower consumed by manual labeling. The category of clothing can be identified in real time to improve the accuracy and speed of clothing retrieval. Therefore, the classification of clothing images has become a research hotspot in the field of machine vision. Lu et al. [1] used part alignment to deal with cross-scenario clothing retrieval. Bossard et al. [2] adopted decision nodes based on multiple random forest learning models to improve the accuracy of clothing identification and classification methods in natural scenes. Simo-sera et al. [3] trained feature extraction networks with sort loss and classification networks with cross-entropy loss to mine the data with weak tags in clothing images. Hidayati et al. [4] proposed a method to identify and classify fashionable women's clothing styles using visually differentiable style elements from full-body images. Yamazaki et al. [5] proposed a feature descriptor focusing on clothing fabrics, wrinkles, and cloth overlaps, and realised a clothing classification method for a single image. Ding et al. [6] used Gaussian blur and visual features based on a scale-invariant feature transform to obtain key points with robustness and representability. Wu et al. [7] adopted a random forest and multi-class support vector machine (SVM) to achieve a fine-grained classification of images of women's fashion wear. However, traditional neural networks and SVMs have the disadvantages of large computing workload and poor real-time performance.

The essence of deep learning [8-10] is to learn more useful features by building machine learning models with many hidden layers and massive training data, so as to improve the accuracy of classification. However, deep networks also exhibit a large computational burden. An extreme learning machine (ELM) is based on a feedforward neural network (FNN) [11-12], which provides a new idea for rapid classification technology. Liang et al. [13] proposed an online sequence extreme learning machine (OSELM) based on ELMs. Zhou [14] proposed an improved whale optimisation algorithm (WOA) to optimises an ELM for classification. In [15], a fabric wrinkle level classification via an improved OSELM was proposed. In [16], the differential evolutionary algorithm was used for iterative optimisation to obtain the input weight and hidden layer bias of a regularised ELM. To improve the generalisation ability of neural networks, a novel neural network called regularised least-squares classification was previously proposed [17].

The dragonfly algorithm (DA) is primarily inspired by the static and dynamic flocking behaviour of dragonflies in nature. More specifically, it is designed by simulating the social interaction of dragonflies observed while they are navigating, finding food, and avoiding enemies. Among the several intelligence optimisation algorithms, the DA is widely used, owing to its simple structure and stable search performance [18].

Inspired by the literature above, this study proposes an OSELM model based on DA optimisation for clothing image classification. The main contributions of this study are as follows:

(1) The input weights and hidden layer bias of the OSELM are randomly generated, and to reduce the impact of random generation on the results, we propose an OSELM model based on the DA. The optimal position of the dragonfly population generated by the DA is used to optimise the in-

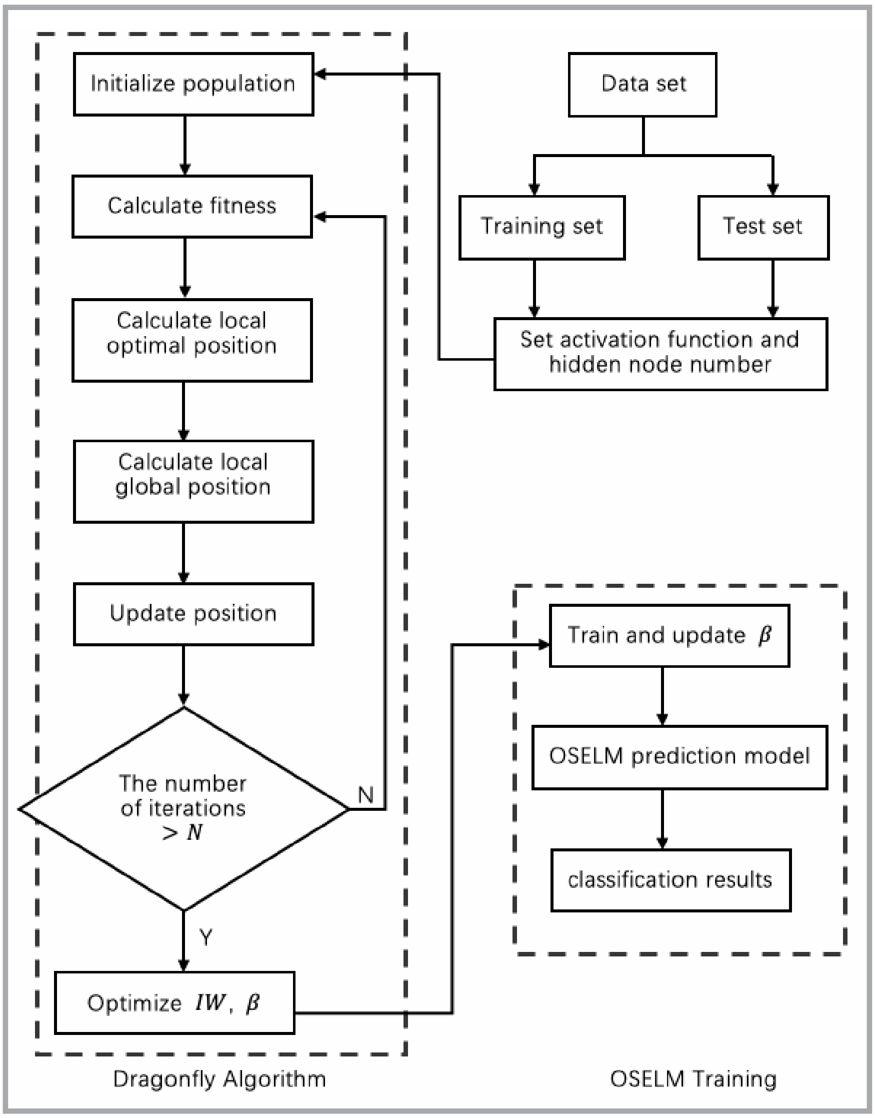

***Figure 1.** Flow chart of DA-OSELM.*

put weight and hidden layer bias of the OSELM to reduce their randomness, and reasonable accuracy results were obtained without human intervention.

(2) We used Fashion-MNIST as the clothing image dataset. This paper compares the classification effect of the differential evolution (DE)-OSELM, DE-ELM, particle swarm optimisation (PSO)-ELM, OSELM, ELM, multilayer (ML)-ELM, and WOA-ELM models. The classification accuracy of the model proposed in this study can reach 93.98% with 350 hidden nodes, which is superior to the classification accuracy of other algorithms with the same number of hidden nodes. Stability analysis of the box graph shows that the model proposed has no outliers.

Related models

Online Sequential Extreme Learning Machine

Unlike the traditional ELM, the OSELM adds a certain number of samples into the model, and the output weights obtained in the previous stage of training can be updated. Such a pattern is regarded as an online sequential model. Using the idea of the ELM, we hoped to find the output weight β to obtain the minimum value of $\left\| H\beta - T \right\|$, where H is the output matrix of the hidden layer, and T is the desired output value. During the initialisation phase, the initial output weight β_0 was obtained as the ELM dose.

$$\beta_0 = K_0^{-1} H_0^T T_0 \qquad (1)$$

Where, $K_0 = H_0^T H_0$. In the online learning phase, where N_1 samples were added to the model, we employed a generalised inverse algorithm to calculate β_1 using the basic idea of ELM:

$$\beta_1 = K_1^{-1} \begin{pmatrix} H_0 \\ H_1 \end{pmatrix}^T \begin{pmatrix} T_0 \\ T_1 \end{pmatrix} \qquad (2)$$

Where, $K_1 = \begin{pmatrix} H_0 \\ H_1 \end{pmatrix}^T \begin{pmatrix} H_0 \\ H_1 \end{pmatrix}$. In online learning, we use β_0 to represent β_1,

$$\begin{pmatrix} H_0 \\ H_1 \end{pmatrix}^T \begin{pmatrix} T_0 \\ T_1 \end{pmatrix} = (K_1 - H_1^T H_1)\, \beta_0 + H_1^T T_1 \qquad (3)$$

and we get

$$\beta_1 = \beta_0 + K_1^{-1} H_1^T (T_1 - H_1 \beta_0) \qquad (4)$$

Further,

$$K_{K+1} = K_K + H_{K+1}^T H_{K+1} \qquad (5)$$

$$\beta_{K+1} = \beta_K + K_{K+1}^{-1} H_{K+1}^T (T_{K+1} - H_{K+1} \beta_0) \qquad (6)$$

K_{K+1}^{-1} can be calculated by the Woodbury formula; choose variable $P_{k+1} = K_{K+1}^{-1}$, and we obtain

$$P_{K+1} = P_K - + P_K H_{K+1}^T (I + H_{K+1} P_K H_{K+1}^T)^{-1} H_{K+1}^T P_K \qquad (7)$$

$$\beta_{K+1} = \beta_K + P_{K+1} H_{K+1}^T (T_{K+1} - H_{K+1} \beta_k) \qquad (8)$$

Dragonfly algorithm

The principle of the DA is to set the initial positions of the dragonfly, its natural enemy, and food, and then update the position of the dragonfly continuously by avoiding the natural enemy and looking for food until the best position is found. The best position is the food, and the worst is the natural enemy. The goal of the algorithm is to get as close to the food and as far away from the natural enemy as possible. The step vector was updated as follows:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + + fF_i + eE_i) + w\Delta X_t \qquad (9)$$

Where, w stands for the inertia weight, S_i, A_i, C_i, F_i and E_i refer to the separation, alignment, cohesion, food factor, and natural enemy factor of the i-th individual, respectively. s, a, c, f, and e refer to the separation weight, alignment weight, cohesion weight, food factor, and natural enemy factor, respectively, and t represents the current iteration number. When $N > 0$,

$$X_{t+1} = X_t + \Delta X_{t+1} \qquad (10)$$

When $N = 0$, the individual goes for a random walk,

$$X_{t+1} = X_t + L\acute{e}vy(d) \times X_t \qquad (11)$$

DA-OSELM classification model proposed

It is common knowledge that in an OSELM, the neuron input weights and hidden layer bias are randomly generated. At the same time, it can be found that these two values directly affect the performance of the output weight and thus have a considerable impact on the result. The DA can help optimise the two parameters of the OSELM. In the original DA, the fitness value was calculated using the objective function. To match the next OSELM classification calculation, the OSELM was used to calculate the corresponding fitness value. The fitness value calculated by multiple iterations of the OSELM helped update the dragonfly position.

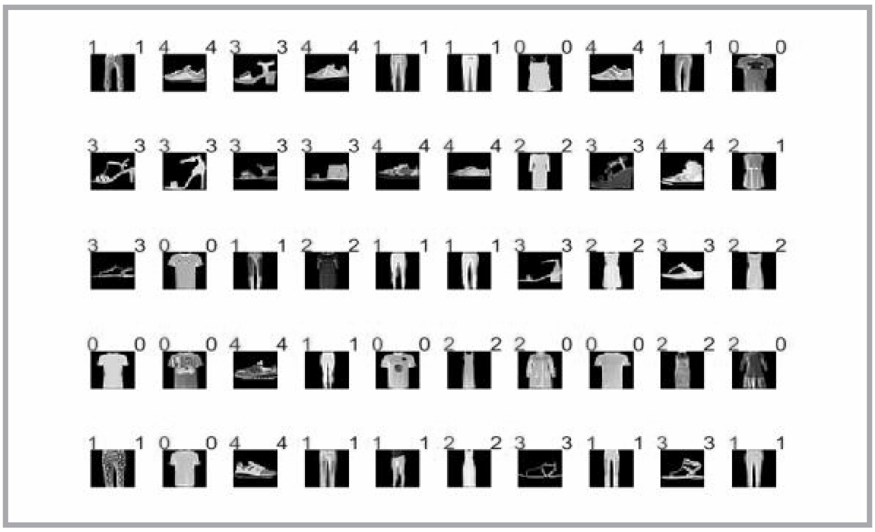

Figure 2. Costume picture results by DA-OSELM.

In the DA, the number of neural network input nodes is N_n, and that of neural network hidden layer nodes is N_h. More importantly, each dragonfly population should have $N_n \times (N_h + 1)$ dragonflies, and we can obtain the optimal location for the dragonflies. In these optimal positions, the $N_n \times N_h$ optimal value is assigned to the input weight and the N_h optimal position to the hidden layer bias. Thus, the optimisation phase of the DA ends. Next, the input weight and hidden layer bias obtained are introduced into the OSELM for calculation. It can be found that the accuracy rate of classification obtained is higher than that of the classification using the OSELM alone. The flow diagram for the DA-OSELM is shown in ***Figure 1***.

Clothing image data set

The data set used in this study was from the Fashion-MNIST dataset [19], which is a clothing image dataset released by Zalando Research, a German organisation. In this study, 10.000 training set samples, 5000 test set samples, and five categories were selected from the Fashion-MNIST dataset, among which there were 2000 training set samples and 1000 test set samples for each category.

Experiment

Algorithm parameters

A comparison with the DA-OSELM model proposed in this paper is presented through the following configurations: models DE-OSELM, DE-ELM [20], PSO-ELM, OSELM [13], ELM [11], ML-ELM [21], and WOA-ELM [14].

In the ELM and OSELM classification models, the number of hidden layer nodes was set to $L = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 100, 150, 200, 250, 300\}$. Moreover, there were different activation functions in these models, g = {'Sigmoid', 'Hardlim', 'Tribas', 'Radbas'. The DA was able to optimise the input weight and hidden layer bias of the OSELM. In this study, we employed the DE, PSO, ML, WOA, and other algorithms to optimise the corresponding ELM for comparative experiments. In the parameter settings of the DA, the set population size is {10, 20, 30, 40, 50}, the maximum iteration number – (10, 50), the upper limit – 1, and the lower limit is – 1. In the parameter settings for the OSELM, the initial input sample number is 100, the block number – 50, the activate function – 'sig', and the hidden neuro number – 50. In the parameter settings for the DE, the population size is 10, the maximum iteration number 20, set F = 1 and CR = 0.8. In the parameter settings for PSO, the population size is 10, the maximum iteration number – 20, the velocity (-1, 1), and set $C_1 = C_2 = 2$. In the parameter settings for WOA, the population size is 10, the maximum iteration number – 20, and A $\in$ [0, 2]. The dataset images in this paper contained all processed grayscale images, and the pixel value ranged from 0 to approximately 255. The colour of the background was black and that of the clothes was white. When processing the input data, all input values were normalised in the range of [0, 1].

Results and analysis

In this paper, the results of the DA-OSELM algorithm proposed are compared with those of the other algorithms to evaluate the effect of the DA-OSELM. The average classification accuracy (Avg), standard deviation (Stdv), best classification accuracy (Best), and worst classification accuracy (Worst) were used to evaluate the performance of the algorithm. ***Tables 1*** and ***2*** list the comparison performance. It can be seen from

Table 1. *Comparison of DA-OSELM, OSELM, ELM, DE-OSELM, and DE-ELM models.*

Measure	DA-OSELM	OSELM	ELM	DE-OSELM	DE-ELM
Avg	**0.9400**	0.9344	0.9338	0.9351	0.9392
Stdv	0.0013	**0.0011**	0.0015	0.0012	0.0014
Best	**0.9422**	0.9356	0.9356	0.9370	0.9410
Worst	**0.9390**	0.9330	0.9320	0.9338	0.9380

Table 2. *Comparison of DA-OSELM, ML-ELM, PSO-ELM and WOA-ELM models.*

Measure	DA-OSELM	ML-ELM	PSO-ELM	WOA-ELM
Avg	**0.9400**	0.9356	0.9380	0.9378
Stdv	**0. 0013**	0.0030	0.0015	0.0015
Best	**0. 9422**	0.9396	0.9402	0.9392
Worst	**0. 9390**	0.9314	0.9364	0.9356

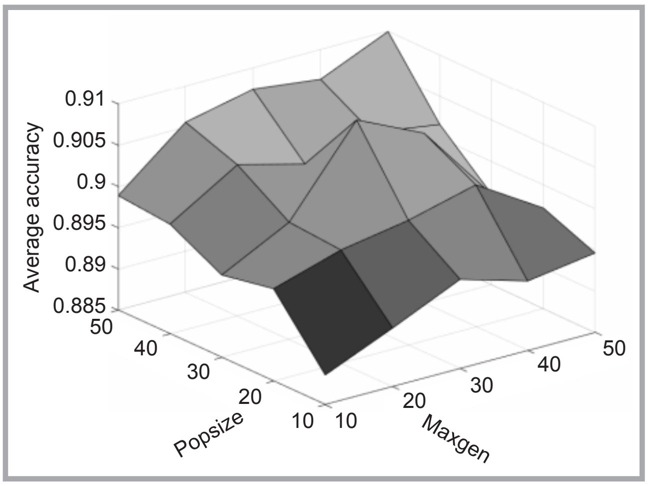

Figure 3. *Variation figure of average accuracy (n = 50).*

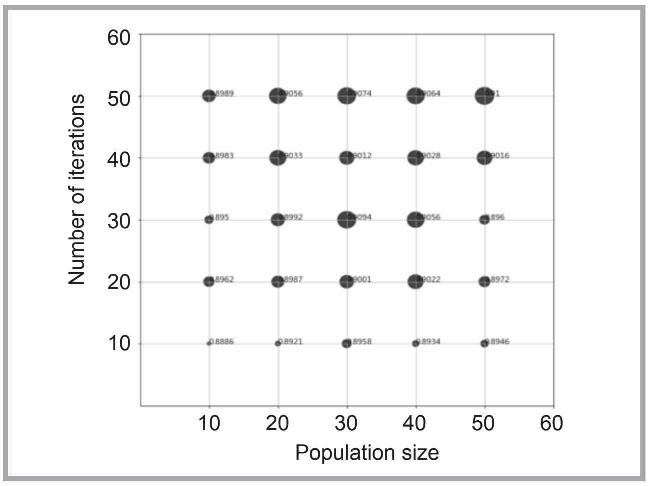

Figure 4. *Bubble diagram (n = 50).*

the Tables that the DA-OSELM has the best average classification accuracy, the best classification accuracy, and the worst classification accuracy. Although the standard deviation is not the best, the algorithm model proposed still has the best effect.

In ***Figure 2***, 50 images from one running result are selected and marked with true labels and predicted labels on the left and right sides respectively. If the training label is predicted correctly for one of the test set samples, the black asterisk is in the blue circle. When the prediction is wrong, the most intuitive manifestation is the error in red. The prediction error was only 0 or 1. If it is 0, the prediction is correct; if it is 1, the prediction is wrong.

In ***Figure 2***, 50 images from a single run were selected and marked with the red true label and blue prediction label on the left and right sides. When the left and right labels were the same, it indicated that the image prediction was correct. It can be intuitively seen whether the dataset images in this study that correspond to the predicted label values and errors that may affect the accuracy due to classification mismatches are excluded.

Sensitivity and stability analysis

Parameters of DA

The parameters of the DA that exhibited the greatest influence on the results are the maximum number of iterations and the population number *m*. To study the influence of these two parameters on the experimental results, control variates were used to combine the maximum iteration number *t*, population number *m*, and the number of nodes in the hidden layer *n* of the OSELM. Two of the three variables were changed each time, while the other was fixed, to observe their influence on classification accuracy. Among them, 5 different values were used for the population number, maximum iteration times, and hidden layer nodes: 10, 20, 30, 40, and 50. When the node of the hidden layer was fixed, 50 was adopted, 20 when the maximum number of iterations was fixed, and 10 when the population number was fixed. The method of expression was a relative three-dimensional graph, which has more intuitive and vivid advantages compared to two-dimensional graphs.

Figure 3 illustrates the corresponding performances when the parameters of the algorithm were changed separately. It can be easily observed that the number of hidden layer nodes, population size, and the maximum number of iterations within the range of (10, 50) gradually improved the classification accuracy. When these three parameters were between 10 and 20, the change in the classification accuracy rate was the most obvious, and then the change decreased gradually. In addition, the range of variations between 40 and 50 was much smaller. This indicates that within the range of (10, 50), these three parameters had the greatest impact on the classification accuracy.

In ***Figure 4***, the bubble size represents the average accuracy. It can be clearly seen from these figures that a higher number of hidden nodes led to proportionately better results. The maximum number of iterations also had a similar influence on the result, with larger iterations leading to better results. However, the effect of population size on the results was not obvious because of fluctuations in the number of different populations.

Table 3. *Average classification accuracy of activation functions.*

Hidden nodes	Sigmoid	Hardlim	Tribas	Radbas
10	0.8928	0.7412	0.5630	0.5762
20	0.8401	0.7859	0.5482	0.5754
30	0.8682	0.8178	0.6192	0.6964
40	0.8805	0.8452	0.5710	0.5350
50	0.8928	0.8330	0.6334	0.6750
60	0.8884	0.8228	0.6698	0.6668
70	0.9038	0.8628	0.6362	0.7654
80	0.9006	0.8532	0.6770	0.7842
90	0.9132	0.8636	0.6570	0.8218
100	0.9142	0.8666	0.7948	0.8052

OSELM activation functions

The calculation results before the input activation function could have been either very positive or very negative. In this case, the activation function needed to process the result, limit the result to a certain interval, and make it close to the desired result. Therefore, the selection of the activation function was very important. This study chose 10 sets of data such that each group data of the hidden layer nodes were different. L = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}. At the same time, each set of data was compared with different activation functions. The maximum number of iterations was 20, and the population size was 10.

As listed in ***Table 3***, the sigmoid activation function was the best, followed by the Hardlim function, Radbas function, and then the Tribas function. The sigmoid and Hardlim functions had a relatively stable influence on the results, and the accuracy also increased with an increase in the number of hidden nodes. However, the results of Radbas and Tribas fluctuated considerably as the number of hidden nodes increased and were significantly less effective than the other two activation functions. Overall analysis shows that the sigmoid function produced the most accurate result and was, therefore, used as the activation function in the DA-OSELM in this study.

Number of hidden nodes

To study the influence of the number of hidden layer nodes on the classification performance of the DA-OSELM, an algorithm simulation was carried out using five neurons, consequently ending with 50 neurons, adding 5 neurons each time.

Figure 5 shows the average accuracy rate of different algorithms at the nodes from 50 to 350 hidden layers. It can be found that the DA-OSELM had an obvious advantage at the nodes from 5 to 100. After that, the accuracy of all algorithms improved, but the DA-OSELM was still the best algorithm.

As seen from ***Table 4***, the accuracy of all algorithms increased gradually from the 50th to the 350th node, until it was almost impossible to improve the accuracy, and some algorithms even declined, instead of rising. After the 100th node, the accuracy of all algorithms had not changed much, but the DA-OSELM still had certain advantages over other algorithms.

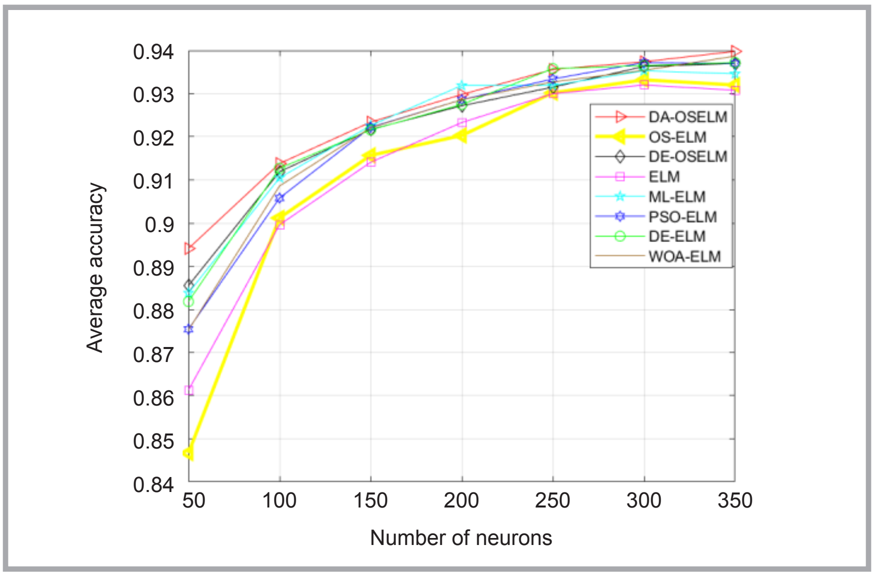

Figure 5. *Average accuracy of different algorithms (n = (50, 350)).*

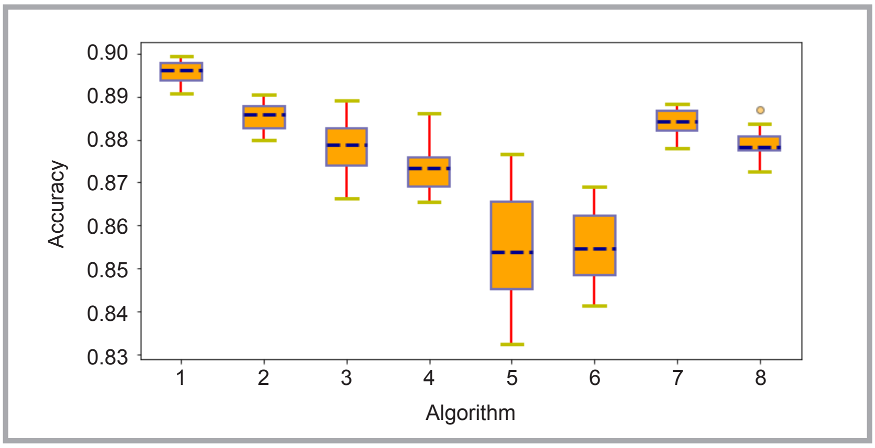

Figure 6. *Box plot for different algorithms (1-8 for DA-OSELM, DE-OSELM, ML-ELM, PSO-KELM, OSELM, ELM, DE-ELM, WOA-ELM).*

Stability

Figure 6 shows the classification accuracy distribution for each algorithm. Each box in the box plot represents the result of the 10-fold operations. The box plot has five data features for each set of data. The top line represents the upper limit, the top of the square – the upper quartile, the black line – the median, the bottom of the square – the lower quartile, and the bottom line represents the lower limit. In addition, there may be a yellow origin representing the outlier. As seen from ***Figure 6***, the stability of the WOA-ELM model was the highest, but the black line was lower, and there were outliers. The DE-OSELM algorithm also had high stability, but the black line was on the higher side, and the data were

Table 4. *Average accuracy of different algorithms (n = (50, 350)).*

Node number	DA-OSELM	OSELM	DE-OSELM	DE-ELM	PSO-ELM	ML-ELM	WOA-ELM	ELM
50	0.8941	0.8467	0.8855	0.8818	0.8754	0.8838	0.8725	0.8613
100	0.9138	0.9012	0.9119	0.9127	0.9058	0.9105	0.9086	0.8996
150	0.9234	0.9157	0.9217	0.9216	0.9222	0.9226	0.9221	0.9141
200	0.9298	0.9203	0.9272	0.9275	0.9286	0.9319	0.9286	0.9233
250	0.9356	0.9302	0.9315	0.9358	0.9334	0.9319	0.9327	0.9300
300	0.9374	0.9332	0.9363	0.9365	0.9372	0.9353	0.9354	0.9320
350	0.9398	0.9320	0.9370	0.9372	0.9370	0.9346	0.9387	0.9308

Table 5. *T-test results.*

Algorithm	Comparison algorithm	H	P
DA-OSELM	OSELM	1	8.1936E-05
	DE-OSELM	1	2.6251E-04
	ELM	1	1.0506E-04
	PSO-ELM	0	0.05829
	ML-ELM	1	0.0160
	DE-ELM	0	0.4201
	WOA-ELM	1	0.0385

not evenly distributed. The data range of other stable algorithms, including the ML-ELM and DE-ELM, was also large. The data of each box plot shows that the distribution accuracy of DA-OSELM was the highest.

Significance analysis

To detect whether there was a significant difference between DA-OSELM and other comparable algorithms, we conducted a T-test. Typically, if the value of H is equal to 1, i.e., the value of P is less than 0.05, then there is a significant difference between the two algorithms. As is evident from ***Table 5***, the DA-OSELM was significantly different from the OSELM, DE-OSELM, ELM, ML-ELM, and WOA-ELM. The P value was close to 0.05 compared to that of the PSO-ELM algorithm, showing a certain difference. The P value was greater than 0.05, as compared to that of the DE-ELM algorithm, showing no significant difference.

Conclusions

To accurately classify clothing images, the Fashion-MNIST dataset was used as the research object in this study, and an online sequential extreme learning machine based on the dragonfly algorithm was proposed for a clothing classification model. To improve the accuracy of the classification results, the dragonfly algorithm was used to optimise the input weight and hidden layer bias. The model proposed in this paper had a classification accuracy of up to 93.98 % using 350 hidden layer nodes. In the stability analysis of the box plot, the algorithm proposed exhibited uniform data distribution and no outliers, whereas the other comparable algorithms had outliers or were less stable than our algorithm.

Acknowledgements

This work was supported by the National Key R&D Program of China (2017YFB1304000).

References

1. Liu S, Song Z, Liu GC, et al. Street-to-shop: Cross-Scenario Clothing Retrieval Via Parts Alignment and Auxiliary Set. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2012; 3330-3337.
2. Bossard L, Dantone M, Leistner C, et al. Apparel Classification with Style. *Proceedings of Asian Conference on Computer Vision* 2012; 321-335.
3. Simo-Serra E, Ishikawa H. Fashion Style in 128 Floats: Joint Ranking and Classification Using Weak Data for Feature Extraction. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* 2016; 298-307.
4. Hidayati SC, You CW, Cheng WH, et al. Learning and Recognition of Clothing Genres from Full-Body Images. *IEEE Transactions on Cybernetics* 2018; 48(5): 1647-1659.
5. Yamazaki K, Inaba M. Clothing Classification Using Image Features Derived from Clothing Fabrics, Wrinkles and Cloth Overlaps. *IEEE International Conference on Intelligent Robots and Systems* 2013; 2710-2717.
6. Ding XJ, Zou CH, Chen JY, et al. Extraction and Classification of the Nationality Clothing via Visual Features. *Textile Research Journal* 2016; 86(12): 1259-1269.
7. Wu MM, Liu L, Fu XD, et al. Fine-Grained Clothing Image Classification by Style Feature Description. *Journal of Computer-Aided Design and Computer Graphics* 2019; 31(5):780-791.
8. Wang SH. COVID-19 Classification By CCSHNet with Deep Fusion using Transfer Learning and Discriminant Correlation Analysis. *Information Fusion*, 2020, DOI: 10.1016/j.inffus.2020.11.005.
9. Zhang YD. A Five-Layer Deep Convolutional Neural Network with Stochastic Pooling for Chest CT-Based COVID-19 Diagnosis. *Machine Vision and Applications* 2021; 32, Article ID: 14.
10. Zhang YD. A Seven-Layer Convolutional Neural Network for Chest CT Based COVID-19 Diagnosis Using Stochastic Pooling. *IEEE Sensors Journal* 2020; DOI: 10.1109/JSEN.2020.3025855.
11. Huang GB, Zhu QY, Siew C. Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. *IEEE International Joint Conference on Neural Networks* 2004; 2: 985-990.
12. Zhang YD, Zhao GH, Sun JD, et al. Smart Pathological Brain Detection by Synthetic Minority Oversampling Technique, Extreme Learning Machine, and Jaya Algorithm. *Multimedia Tools and Applications* 2018; 77(17): 22629-22648.
13. Liang NY, Huang GB, Saratchandran P, et al. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Trans Neural Netw* 2006; 17:1411-1423.
14. Zhou ZY, Wang C, Zhang JX, et al. Color Difference Classification of Solid Color Printing and Dyeing Products Based on Optimization of the Extreme Learning Machine of the Improved Whale Optimization Algorithm. *Textile Research Journal* 2020; 90(2): 135-155.
15. Zhou ZY, Zhang RX, Zhang JX, et al. Fabric Wrinkle Level Classification via Online Sequential Extreme Learning Machine Based on Improved Sine Cosine Algorithm. *Textile Research Journal* 2020; 90(17-18): 2007-2021.
16. Zhou Z, Wang C, Gao X, Zhu Z, Hu X, Zheng X, Jiang L. Fabric Defect Detection and Classifier via Multi-Scale Dictionary Learning and an Adaptive Differential Evolution Optimized Regularization Extreme Learning Machine. *FIBRES & TEXTILES in Eastern Europe* 2019; 27, 1(133): 67-77. DOI: 10.5604/01.3001.0012.7510.
17. Gan HT, She QS, Ma YL, et al. Generalization Improvement for Regularized Least Squares Classification. *Neural Computing & Applications* 2019; 31 (s2): 1045-1051.
18. Mirjalili S. Dragonfly Algorithm: A New Meta-Heuristic Optimization Technique For Solving Single-Objective, Discrete, and Multi-Objective Problems. *Neural Computing & Applications* 2016; 27(4):1053-1073.
19. Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747.
20. Zhu QY, Qin AK, Suganthan PN, et al. Evolutionary Extreme Learning Machine. *Pattern Recognition* 2005; 38:1759-1763.
21. Kasun C, Zhou H, Huang G, et al. Representational Learning with Elms for Big Data. *IEEE Intelligent Systems* 2013; 28(6):31-34.

Received 06.11.2020 Reviewed 04.02.2021