Scientific Reports

https://doi.org/10.1038/s41598-026-44876-9

Article in Press

Ancient architecture image classification with progressive stacking pseudoinverse learning

Received: 20 February 2026

Accepted: 16 March 2026

Published online: 23 March 2026

Cite this article as: Cai Z., Sun X., Zhang S. *et al.* Ancient architecture image classification with progressive stacking pseudoinverse learning. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-44876-9

Zhenjiao Cai, Xuejian Sun, Sulan Zhang, Zhicheng Zhao & Peng Wu

Ancient Architecture Image Classification with Progressive Stacking Pseudoinverse Learning

Zhenjiao Cai[1*], Xuejian Sun[2†], Sulan Zhang[3†], Zhicheng Zhao[4†], Peng Wu[1†]

[1*]Department of Computer Engineering, Taiyuan Institute of Technology, No. 31, Xinlan Road, Taiyuan, 030008, China.
[2]Network and Information Center, Taiyuan Institute of Technology, No. 31, Xinlan Road, Taiyuan, 030008, China.
[3]School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, 030024, China.
[4]Taiyuan Institute of Technology, No. 31, Xinlan Road, Taiyuan, 030008, China.

*Corresponding author(s). E-mail(s): caizj224@163.com;
Contributing authors: 18435155779@163.com; zhsulan@126.com; zhzhich@126.com; 14112078@bjtu.edu.cn;
†These authors contributed equally to this work.

Abstract

In the task of ancient architecture image classification, due to the unconstrained random initialization of the weight matrix, the existing random projection branches cannot effectively focus on the key feature dimensions such as roof contour and bucket arch structure. This not only weakens the core basis for classification but also impairs the ability to distinguish between similar building variants. Meanwhile, the training mode of inputting all samples at one time is easy to cause the model to over-fit the architectural style of specific regions and weaken the cross-regional generalization performance. To address these issues, we propose an ancient architecture image classification with progressive stacking pseudoinverse learning (AAPSP). It comprises two core modules: the key features stacking pseudoinverse learning of ancient architecture (KFSP) and the progressive optimization learning model of ancient architecture samples (POL). Specifically, KFSP initializes the weight matrix to a specific pattern that conforms to the distribution of architectural features (such as Gaussian distribution focusing on the roof contour, uniform distribution capturing decorative patterns).

By combining the attention mechanism (AM), a higher weight is assigned to the base learner of key component recognition to optimize the feature space modeling. POL employs a dynamic sample screening strategy to preferentially select sample iterative optimization models with rare features to reduce redundancy and enhance generalization ability. Experiments on six types of Chinese ancient architecture datasets show that AAPSP performs well in accuracy, precision, recall rate and F1 score.

Keywords: Ancient architecture image classification, stacking pseudoinverse learning, key features, sample progressive optimization

1 Introduction

Chinese ancient architecture carries thousands of years of construction skills and cultural genes [1, 2]. Its core features such as roof shape, bracket structure and decorative patterns have distinct regional and contemporary imprints. It is an important object of digital protection of cultural heritage. With the development of ancient building surveying and mapping technology and image acquisition methods, the image data of various temples, palaces and other buildings are growing exponentially. It is difficult to meet the efficient management needs of large-scale image resources only by relying on the traditional way of classifying ancient building experts through shape comparison. As the core branch of computer vision, image classification technology [3–6] provides a technical possibility for the automatic classification of ancient building images. Its research has irreplaceable value for the construction of digital archives of ancient buildings, auxiliary architectural style dating and rescue protection of cultural heritage.

In the field of image classification, the deep learning algorithm [7–9] relies on the autonomous learning ability of high-dimensional features. In the image classification task with complex texture features such as ancient buildings, the accuracy is significantly better than the traditional manual feature extraction method. However, the mainstream deep learning model relies on the gradient descent optimization strategy [10], In the image classification of ancient buildings, it often faces problems such as slow convergence and easy to fall into local optimum. Especially when there are subtle differences such as similar roof curves and similar bucket arch variants in the sample, the gradient update is easily disturbed by noise. To break through this limitation, non-gradient descent algorithm has become an important supplement. Pseudoinverse Learner (PIL) [11, 12] effectively shortens the training period and avoids the problem of gradient disappearance with one-step non-iterative learning characteristics. Nevertheless, due to the limited abstract ability of ancient architectural features, the classification accuracy and generalization performance are insufficient. In contrast, stacked generalization [13–17] uses the collaborative decision-making of multiple base learners. It can capture multi-scale feature associations such as bucket arch combination and ridge beast arrangement in ancient building images, and significantly

improve classification robustness. Yet, the complex layered structure makes it computationally expensive to process large-scale temple images. The stacking pseudoinverse learner (SP) [14] integrates the advantages of both, retaining the characteristic of fast convergence while improving accuracy, thus providing a new approach for ancient architecture image classification.

As an ensemble learning framework, the performance of the SP is highly dependent on its capacity to model the feature subspace of ancient buildings. The random projection branch [18–23] constructs a multi-view feature space via feature mapping. However, existing methods employ unconstrained randomly initialized weight matrices, which struggle to focus on the critical feature dimensions of ancient buildings. For instance, the roof slope and wall color are placed in the same feature weight, which will dilute the core classification basis such as bucket arch, resulting in a decrease in the ability of the model to distinguish similar building variants. At the same time, the training mode of inputting all samples at one time is easy to make the model overfit the decorative style of specific regional buildings, resulting in classifier redundancy and weakening the generalization ability of cross-regional ancient buildings.

Aiming at the particularity of ancient building image classification, we propose an ancient architecture image classification with progressive stacking pseudoinverse learning (AAPSP), which is used to optimize the capture and utilization efficiency of ancient building features. The framework consists of two core modules:

(1) Key features SP of ancient architecture (KFSP): Aiming at the hierarchy of ancient architectural features, the input weight matrix is initialized to a specific pattern that conforms to the distribution of architectural features (such as focusing on roof contour features with Gaussian distribution and capturing decorative patterns with uniform distribution). Through the AM [24–27], a higher weight is assigned to the base learner that identifies key components such as bucket arches and ridge beasts, so that the stacked feature space is more suitable for the shape logic of ancient buildings, and the classification accuracy and generalization performance are improved.

(2) Progressive optimization learning model of ancient architecture samples (POL): Considering that there are a large number of redundant images with high similarity in ancient architecture samples (such as the same palace facade taken at an angle). Based on Active Learning (AL) [28, 29], we adopt a dynamic sample selection strategy-dividing the data set into a training set and a candidate set, and preferentially select a sample iterative optimization model containing a rare bucket arch shape and a special roof combination through uncertain sampling. On the premise of not adding new data, strengthen the learning of ancient building feature variants and reduce sample selection bias.

Experiments on six datasets of Chinese ancient architecture show that AAPSP can effectively identify the iconic features of ancient buildings and significantly improve the classification accuracy and generalization performance.

The core contributions of this paper are as follows:

(1) The KFSP algorithm is proposed to strengthen the model's ability to capture the characteristics of key components of ancient buildings through feature distribution constraints and attention mechanisms, and to solve the problem of feature weight mismatch in random projection.

(2) The POL dynamic learning mechanism is constructed. Aiming at the contradiction between redundancy and scarcity of ancient building samples, the model is gradually optimized by actively screening samples to reduce the risk of over-fitting.

(3) An AAPSP integration framework for ancient building classification is proposed, which deeply integrates the feature modeling ability of KFSP with the sample optimization strategy of POL to achieve the synergistic improvement of classification accuracy and generalization performance of ancient building images.

(4) The empirical research on six typical ancient building datasets verifies the ability of the proposed method to distinguish different architectural styles and component features, and provides a new paradigm for the intelligent classification of ancient building images.

The remaining organization of this paper is as follows. Section 2 introduces the related research work of this paper, namely PIL, SP and AL. The algorithm of this paper is introduced in detail in section 3. The experimental analysis is carried out in section 4. Finally, section 5 summarizes the full text.

2 Related work

The research of this paper is based on PIL, SP and AL. Therefore, we briefly review these methods.

2.1 PIL

To overcome the shortcomings of BP algorithm, Guo et al. proposed PIL algorithm to train single hidden layer neural network (SHLN) [11]. In this work, the activation function is considered as a hyperbolic function Tanh $(\cdot)$. Minimize the error function in Equation (1) to find the weight parameter matrix W.

$$minimize\|YW - B\|^2 \tag{1}$$

Where $Y = Tanh(XV)$ is the output matrix of the hidden layer, X is the input matrix composed of N input vectors as its rows, $d = n+1$ column as the input vector dimension $n+1$, $B = ArcTanh(T)$, T is the target label matrix, composed of N label vectors as rows, m columns as the target vector dimension.

Equation (1) is a least squares problem, which can be solved by linear algebra. The formal solution of W in Equation (1) is $W = Y^+B$, where Y^+ is the pseudoinverse of Y. It can also be written as:

$$YW - B = YY^+B - B = 0 \tag{2}$$

If Equation (2) holds, then $YY^+ = I$. In the works of Guo et al. [11], the number of hidden layer neurons was set to N for the exact learning purpose.

Thus, a non-iterative fast learning algorithm PIL is obtained. In most cases, only one step is needed to obtain the best solution.

2.2 SP

At present, a variety of studies have been carried out on SP, such as Guo et al. [14] implemented basic SP. They used software reliability growth data to study stack generalization performance. It shows that for noisy data, using stack generalization cannot improve network performance when over-trained network joins. Using a properly trained network, stack generalization can improve network generalization performance. In 2004, Guo et al. [15] also developed a supervised learning algorithm for feedforward neural networks, PIL, which was tested by a case study of the stack generalization application of software reliability growth data. The results show that the proposed algorithm is very effective for the research of computationally intensive generalization technology. In 2018, Guo et al. [16] proposed a hierarchical model for pulsar candidate selection, and assembled a set of trained basic classifiers. The model not only has the advantages of high steady-state accuracy, but also has good generalization performance. For multi-layer perceptron, Feng et al. [17] proposed an integrated model for error modeling. The model contains three base learners, namely a single hidden layer neural network trained by the PIL algorithm, the second is an AE model with L2 loss, both AE and classifier are trained by the PIL algorithm, the last base learner is an AE model with L1 loss trained by CD, and the classifier is trained by the PIL algorithm. By adopting the idea of divide and conquer to adapt to different types of noise, the results of all base learners will enter the meta-learner for further learning, and finally get better performance classification results.

However, these methods obtain the base learner by randomly generating the input weight matrix, ignoring the influence of the distribution of the random projection block itself on the neural network.

2.3 AL

AL overcomes the limitation of limited availability of training samples by iteratively selecting useful samples as training candidates [28]. The most commonly used sample selection methods in AL are random selection [30], mutual information [31], disconnection [32], modified disconnection [33], uncertain sampling [34, 35], committee query (QBC) [36], Fisher information ratio [37], ambiguit [28], and sample selection based on spectral angle mapper and ambiguity (FSAM) [38]. These sample selection methods are classified according to whether the information used to select the sample is spectral or spatial information. The latter depends entirely on the spatial distribution of potential candidates.

But there is not much research on integrating spatial constraints into the AL method [39–43]. Moreover, due to the iterative training of the classifier for each new candidate, the computational complexity of these methods is very high [41]. To solve the above problems, the batch mode AL method has been proposed, which takes into account the uncertainty and diversity of the new selected samples [44]. Work [45] emphasized the benefits of integrating spatial context information into AL, but it did not consider the spatial distribution of newly selected samples. Nevertheless, this method was later extended in [31] to take into account the spatial position of the newly selected sample in the feature space. This conversion will lead to the phenomenon of

point-by-point dispersion in the spatial domain, resulting in multiple visits to the same geographical location [46]. Therefore, the combination of spatial AL achieves better performance than its pixel-level counterparts, and represents a novel and promising research contribution, which has not been fully explored [46].

In 2020, Ahmad et al. [29] proposed an AL method of spatial prior generalized fuzzy extreme learning machine autoencoder (GFELM-AE) to reduce sample selection bias and maintain data stability in spatial domain. GFELM-AE is different from the standard AL method in several aspects. GFELM-AE uses the ambiguity related to the confidence of the training model to classify unknown samples [38], rather than using uncertainty measures. Subsequently, GFELM-AE combines the fuzziness of samples with their diversity to select new training samples with low similarity to existing training samples. Finally, GFELM-AE maintains the balance of the selected sample pool and provides the same representation for all classes to a certain extent, which is achieved by softening the threshold at runtime. The classifier trained on such samples provides better generalization performance on the initially misclassified samples.

3 AAPSP

In this section, we elaborate on the proposed algorithm AAPSP. Firstly, KFSP, one of the core models of AAPSP, is introduced. It realizes the constraint modeling of random projection by assigning specific distribution (such as Gaussian distribution and uniform distribution) to the input weight matrix, so as to accurately focus on the key feature dimensions of ancient buildings. On this basis, the POL is further designed. The model realizes progressive SP by iteratively screening samples with high-value information (such as rare bucket arch and special roof combination samples) to optimize the generalization performance of the model. The specific structure of the AAPSP algorithm is shown in Fig. 1.

3.1 KFSP

The core function of KFSP is to optimize feature modeling and solve the problem of random projection weight mismatch and key features being diluted. The implementation of the model mainly includes three sub-steps: base learner acquisition (weight initialization with feature distribution constraints), AM (key feature weighting), and single hidden layer SP (feature space construction). The framework is shown in Fig. 2.

3.1.1 Acquisition of base learner

Multiple base learners process input samples in parallel, and the weight matrix of each base learner is initialized to a specific distribution to match the characteristics of ancient buildings. In this paper, we generate three V_i with different specific distributions. Where V_1 and V_2 are Gaussian distributions to focus on the roof profile (continuous curve feature). They are subject to the expected value μ_1, the standard deviation δ_1, and the expected value μ_2, the standard deviation δ_2, respectively, V_3 obeys a uniform distribution in the interval [p, q] to capture decortive patterns

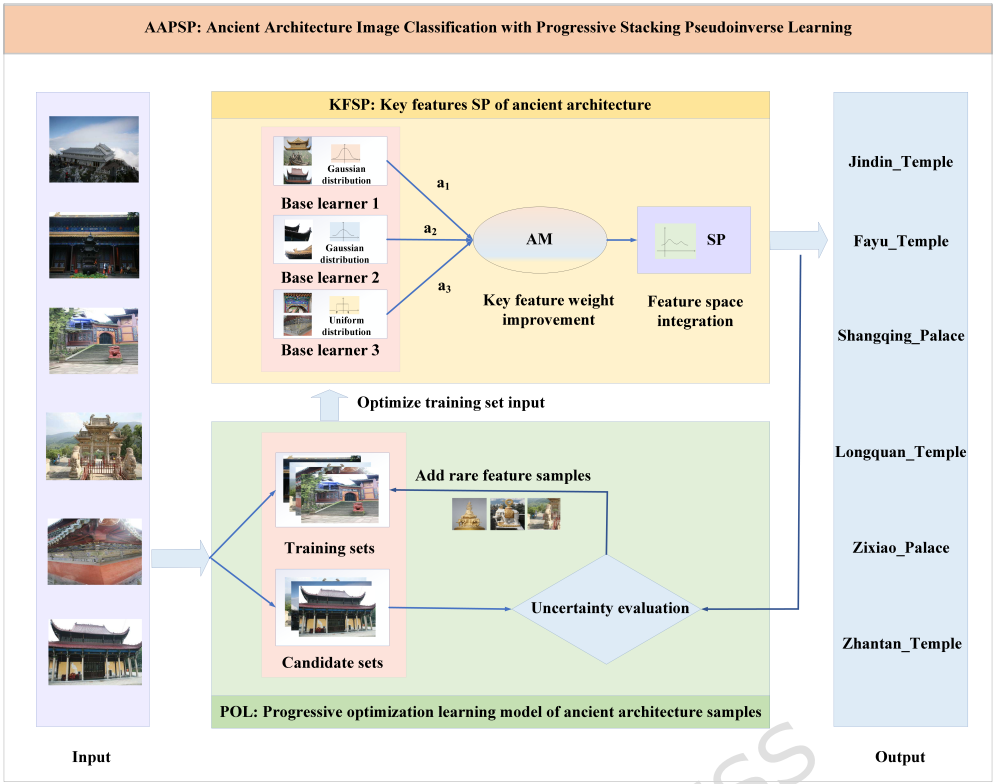


Fig. 1 AAPSP algorithm framework

(discrete texture features):

$$V_1 \sim N(x \mid \mu_1, \delta_1), V_2 \sim N(x \mid \mu_2, \delta_2), V_3 \sim U(p, q) \tag{3}$$

where N represents Gaussian distribution and U is uniform distribution.

At this time, the hidden layer output H_i of the base learner is expressed as:

$$H_i = \sigma(X \times V_i) \tag{4}$$

where, $i = 1, 2, 3$, X is the input data and σ is the Sigmoid activation function.

By the PIL algorithm, the output weight matrix W_i can be calculated:

$$W_i = H_i^{+} \times \text{arc}(\sigma)[T] \tag{5}$$

where, $i = 1, 2, 3$, $^{+}$ represents the pseudoinverse operation, $\text{arc}(\sigma)$ is the inverse activation function, and T is the target label matrix.

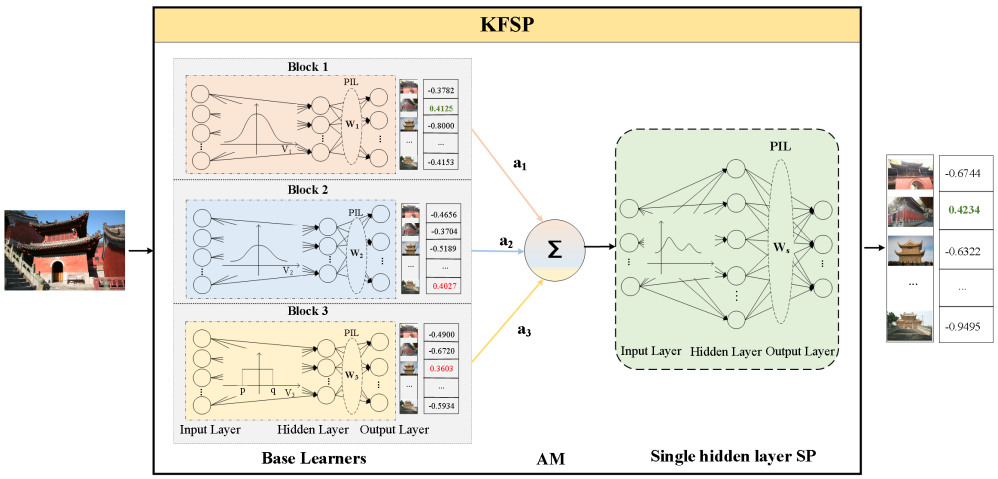


Fig. 2 KFSP algorithm framework

At this point, the network output Y_i of base Learner is expressed as:

$$Y_i = H \times W_i \tag{6}$$

where, $i = 1, 2, 3$.

To verify the effectiveness of the aforementioned distribution constraint strategy, this study conducts a visualization analysis of the feature responses of the three base learners (see Fig. 3). A typical ancient architecture sample from the dataset is selected, and heatmaps of the hidden layer outputs are plotted to intuitively demonstrate the feature focusing effect. For the base learners V_1/V_2 initialized with Gaussian distribution, the high-response regions (orange) accurately cover the roof contour area, while the response to non-roof regions is extremely weak (predominantly blue), with a response ratio of 78.2%±3.5%. Notably, the response intensity of V_2 is slightly higher than that of V_1, further enhancing the capture of roof contour features. For the base learner V_3 initialized with uniform distribution, the feature responses are evenly distributed without obvious extremely low-response regions, and the response ratio to discrete texture features such as decorative patterns reaches 72.6%±4.1%, verifying the adaptability between distribution constraints and target features.

3.1.2 AM

In this section, we introduce the idea of AM to assign higher weights to the base learners that identify key components such as bucket arches and ridge beasts and weaken the influence of secondary features (such as wall color). The main processes are as follows:

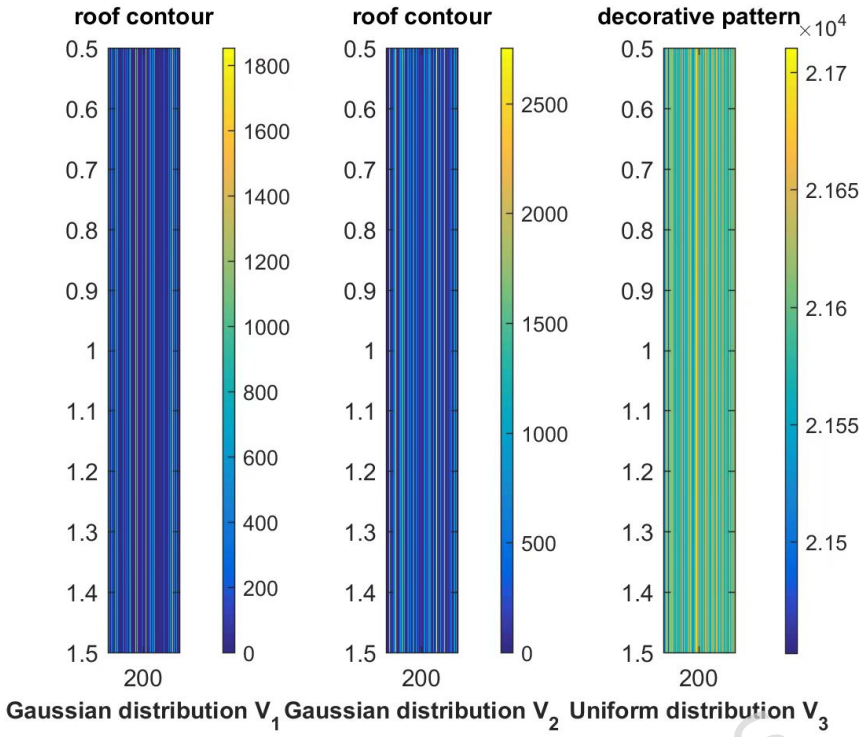


Fig. 3 Feature response heatmaps of KFSP base learners. Left: Focused response of Gaussian-distributed V_1 to roof contour; Middle: Focused response of Gaussian-distributed V_2 to roof contour; Right: Uniform response of uniformly-distributed V_3 to decorative patterns. Note: Orange indicates high feature response regions, and blue indicates low feature response regions. The sample is a typical one from the six categories of ancient architecture datasets.

Firstly, the similarity S_i between the output Y_i of three different random feature distributions and the real label T is calculated, that is:

$$S_i = \frac{Y_i \times T}{\|Y_i\| \times \|T\|} \tag{7}$$

where, $i = 1, 2, 3$.

After the nonlinear mapping of Equation (8) Sigmoid and the normalization of Equation (9) Softmax, a new input weight matrix a_i can be obtained.

$$a_i' = \frac{1}{1 + e^{-s_i}} \tag{8}$$

$$a_i = \frac{e^{a_i'}}{\sum_{j=1}^{3} e^{a_i'}} \tag{9}$$

where, $i = 1, 2, 3$.

3.1.3 Single hidden layer SP

The single hidden layer SP constructs a feature space that fits the ancient architectural form logic by integrating the weighted base learner output, and outputs the classification results (such as temple, palace and other categories). Specifically, according to the new input weight matrix a_i and the network output Y_i of the base learner, the hidden layer output H_s after the stack is :

$$H_s = \sigma(tempH) \tag{10}$$

where, σ is the Sigmoid activation function, and then the output weight matrix W_s after the stack is obtained by the PIL algorithm:

$$W_s = H_s^+ \times arc(\sigma)[T] \tag{11}$$

where $^+$ is the pseudoinverse operation, T is the actual label matrix, and $arc(\sigma)$ is the activation function.

Finally, we obtain the output Y_s of the meta learner:

$$Y_s = H_s \times W_s \tag{12}$$

3.2 POL

The core function of POL is to dynamically screen samples to solve the problem of over-fitting and weak generalization ability caused by redundant samples. It includes three sub-steps: data set initialization division, uncertainty sampling (rare feature screening) and sample iterative update.

3.2.1 Initialization partition of dataset

In this section, we divide the training samples into the training set X_{train} and the candidate set $X_{candidate}$, that is, the initial sample and the sample to be screened. Specifically, we randomly select 20% of the images as the test set, 80% of the remaining samples are divided into the training set, and 20% as the candidate set.

3.2.2 Uncertainty sampling

Based on the classification results of KFSP, the uncertain samples in the candidate set are calculated, such as the samples with low prediction probability and high feature rarity, and the samples with rare features are preferentially screened. Specifically, according to the initialization weight matrix obtained by X_{train} training KFSP, the meta learner of training $X_{candidate}$ to obtain the candidate set outputs $Y_{s\_candidate}$.

3.2.3 Sample iterative update

We perform sample selection on $Y_{s\_candidate}$, move the selected rare feature sample X_f from the candidate set to the training set, and update the training set. Iteration is repeated until the training set covers enough feature variants or reaches the number of

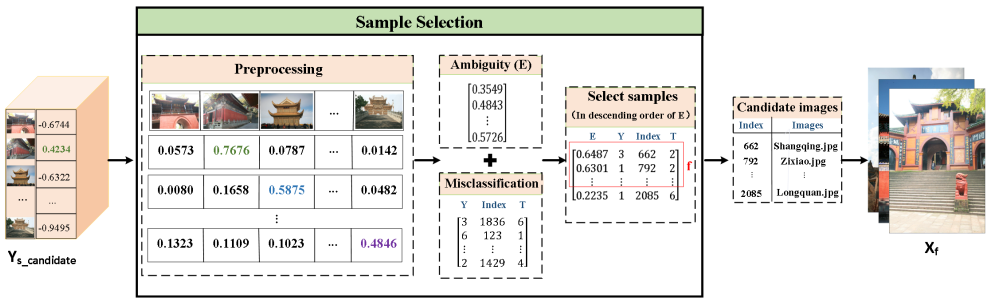


Fig. 4 Sample selection

iterations, and finally the classification results with high generalization performance are obtained.

In fact, our proposed framework can be combined with any sample selection strategy. This paper introduces a spatial prior generalized fuzzy sample selection strategy [22, 23], which can not only maintain the stability of spatial domain data, but also reduce the sample selection bias. By comprehensively considering the fuzziness and diversity of the samples, the misclassified samples with low similarity to the training samples are selected. The specific process is shown in Fig. 4, which mainly includes three steps: preprocessing, ambiguity calculation and sample screening.

(1) Preprocessing

For the meta learner output results of C categories and G candidate samples $Y_{s\_candidate_{ij}}(where\ i \in 1, 2, \ldots, C,\ j \in 1, 2, \ldots, G)$, we first extract the index m corresponding to the maximum value of each column in the $Y_{s\_candidate}$ matrix and retain it. On this basis, to simplify the calculation process of the class edge probability distribution matrix, it is necessary to perform a preprocessing operation on $Y_{s\_candidate}$: it is transformed into a randomization result with values greater than 0 by Equation (13).

$$Y_{s\_candidate_{ij}} = \frac{rand(C, 1)}{C} \tag{13}$$

where C denotes the number of classes, $rand(C, 1)$ is a random vector of C rows and 1 columns.

Then, the mth class of each candidate set is reassigned by Equation (14) to obtain the class edge probability distribution matrix:

$$Y_{s\_candidate_{mj}} = 1 - \sum_{i=1, i \neq m}^{C} Y_{s\_candidate_{ij}} \tag{14}$$

where $i \in 1, 2, \ldots, C$, C is the number of categories, $j \in 1, 2, \ldots, G$, G is the number of candidate samples, m is the index value corresponding to the maximum value in each column of the meta learner output $Y_{s\_candidate}$. $Y_{s\_candidate_{mj}}$ is the mth row and jth

column of $Y_{s\_candidate}$. Σ is a summation operation. $\sum_{i=1,i\neq m}^{C} Y_{s\_candidate_{ij}}$ represents the sum of column j and the remaining $(C-1)$ rows except m.

In this case, if the category is a real category, its marginal probability will approach 1. At the same time, the marginal probability matrix needs to satisfy the properties defined by Equation (15) and Equation (16) [22], as follows:

$$\sum_{i=1}^{C} Y_{s\_candidate_{ij}} = 1 \tag{15}$$

$$0 < \sum_{j=1}^{G} Y_{s\_candidate_{ij}} < G \tag{16}$$

(2) Ambiguity calculation

The mathematical definition of the ambiguity $\mathrm{E}(Y_{s\_candidate})$ of the membership matrix $Y_{s\_candidate}$ (corresponding to G candidate sets of class C) is shown in Equation (17). The definition needs to strictly follow the constraints specified in Equation (15) and Equation (16).

$$\begin{aligned} E(Y_{\text{s\_candidate}}) = \frac{-1}{G \times C} \sum_{i=1}^{C} \sum_{j=1}^{G} \Big[& Y_{s\_candidate_{ij}} \log(Y_{s\_candidate_{ij}}) \\ & + (1 - Y_{s\_candidate_{ij}}) \log(1 - Y_{s\_candidate_{ij}}) \Big] \end{aligned} \tag{17}$$

(3) Sample screening

By constructing the correlation matrix $A = [E(Y_{s\_candidate}), Y_{s\_candidate}, T_{candidate}]$, the ambiguity index $E(Y_{s\_candidate})$, the prediction class label $Y_{s\_candidate}$ and the actual class label $T_{candidate}$ are ternary correlated. Then, based on the ambiguity value, the matrix A is arranged in descending order, and the error classification samples satisfying the condition $Y_{s\_candidate} \neq T_{candidate}$ are selected. On this basis, the top f samples with the highest ambiguity X_f are extracted to form an enhanced subset, which is incorporated into the original training set X_{train}. By doing this, the updated training set and candidate set can be represented as:

$$X_{train} = [X_{train}, X_f], \quad X_{candidate} = X_{candidate} - X_f \tag{18}$$

4 Results and discussion

The experimental environment is the i9-9900X processor, Dell GeForce RTX 2080TI graphics card computer.

4.1 Data sets and implementation details

The data set used in this experiment is a set of 6 types of Chinese ancient architecture images provided by the State Key Laboratory of Pattern Recognition, Institute

of Automation, Chinese Academy of Sciences [47, 48] (as shown in Fig. 5), including Shangqing_Palace, Zixiao_Palace, Jindin_Temple, Zhantan_Temple, Fayu_Temple and Longquan_Temple. A total of 2269 images. This dataset is constructed purely by Gao's team [https://weigao-3dv.github.io/], which can be downloaded from https://doi.org/10.57760/sciencedb.ai.00008 by contacting the owner as indicated in the web site.

Image preprocessing operations were performed on all high-resolution images before inputting into the network, including three key steps: (1) Proportional downsampling was conducted to uniformly adjust the image resolution to 224×224, where the original aspect ratio of the images was strictly maintained and zero-padding was applied to the edges, aiming to reduce computational complexity and adapt to the input dimension of the model; (2) The downsampled images were normalized to scale all pixel values to the range of [0,1], which eliminates the adverse impact of pixel value magnitude differences on model training convergence and feature learning; (3) No additional manual feature extraction operations were conducted, and the model independently learned the deep features of ancient building images to ensure the objectivity and authenticity of feature learning. The division rules of the data set are as follows: 20% of the images are randomly selected as the test set, 80% of the remaining samples are divided into the training set, and 20% as the candidate set. The specific distribution is shown in Table 1.

Table 1 Dataset Details. The table lists the total number of six categories of Chinese ancient architectural images, fixed resolution, and the splitting of training, candidate, and test samples.

Datasets	Number	Sizes	Train	Candidate	Test
Shangqing_Palace	324	4368×2912	207	52	65
Zixiao_Palace	636	4368×2912	407	102	127
Jindin_Temple	418	4368×2912	267	67	84
Zhantan_Temple	158	4368×2912	101	25	32
Fayu_Temple	290	4368×2912	186	46	58
Longquan_Temple	443	4368×2912	284	71	88
Total	2269	-	1452	363	454

To weaken the interference of experimental randomness and accidental factors on the results, all experiments were repeated 10 times independently, and the average classification accuracy (Mean) and standard deviation (STD) were used as performance evaluation indicators. The model parameters are configured as follows: the number of hidden layer neurons is set to 400, the base learner uses ReLU as the activation function, and the stack layer uses Sigmoid as the activation function.

4.2 Evaluating indicator

In this paper, four evaluation indicators are used to quantitatively evaluate the performance of the model, including Accuracy, Precision (P), Recall (R) and F1 Score.

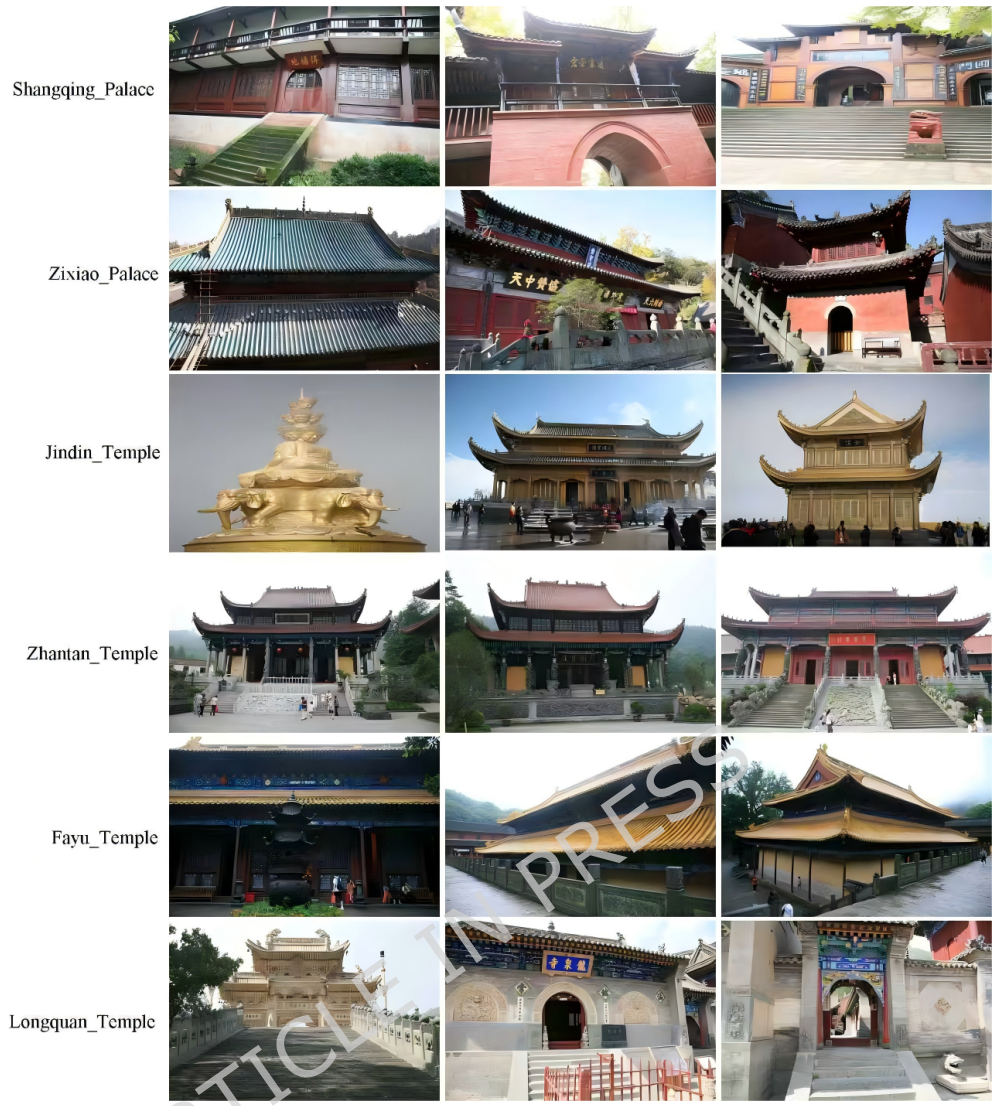


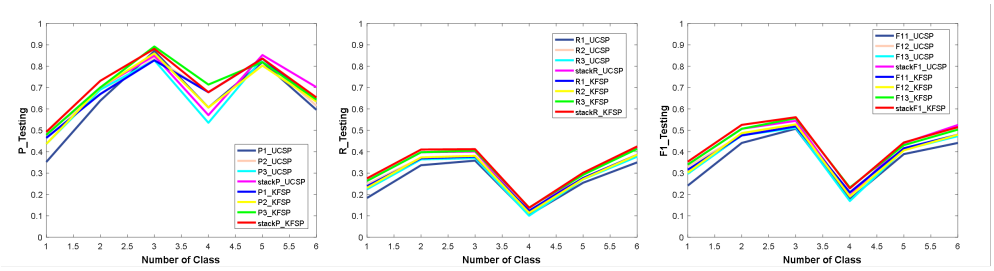
Fig. 5 Ancient architecture image example

The specific calculation formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

$$P = \frac{TP}{TP + FP} \tag{20}$$

$$R = \frac{TP}{TP + FN} \tag{21}$$



(a) P **(b) R** **(c) F1**

Fig. 6 Evaluation of KFSP algorithm

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{22}$$

where TP represents the true case (the actual positive case and the number of samples correctly predicted as positive cases), TN is the true negative case (the actual negative case and the number of samples correctly predicted as negative cases), FP represents the false positive case (the actual negative case and the number of samples incorrectly predicted as positive cases), FN is the false negative case (the actual positive case and the number of samples incorrectly predicted as negative cases).

4.3 Comparison of P, R and F1

4.3.1 Evaluation of KFSP algorithm

In this section, the P, R and F1 score of KFSP algorithm and unconstrained SP (UCSP) before and after stacking are compared and analyzed on the ancient building dataset. Among them, UCSP uses a random initialization method without specific distribution constraints to generate an input weight matrix. The experiment aims to verify the effectiveness of the proposed stack model through two aspects, one of which is to compare the results before and after the stack. The second is to verify the effectiveness of KFSP on the specific distribution constraints of the input weight matrix by the performance difference with UCSP. The specific results are shown in Fig. 6.

From Fig. 6, it can be seen that the scores of P, R and F1 after stacking are slightly higher than the independent results of the three base learners, which confirms the effectiveness of integrating base learners with different specific distributions. Further comparison shows that the indexes of KFSP after stacking are significantly better than those of UCSP. This is because KFSP imposes specific distribution constraints on random features, so that the new feature space generated after stacking has stronger adaptability to data distribution, thus effectively improving the classification performance of the model.

4.3.2 Evaluation of AAPSP algorithm

This section shows the P, R and F1 scores of the proposed algorithm on the ancient architecture dataset, as shown in Fig. 7. The experimental results show that the P,

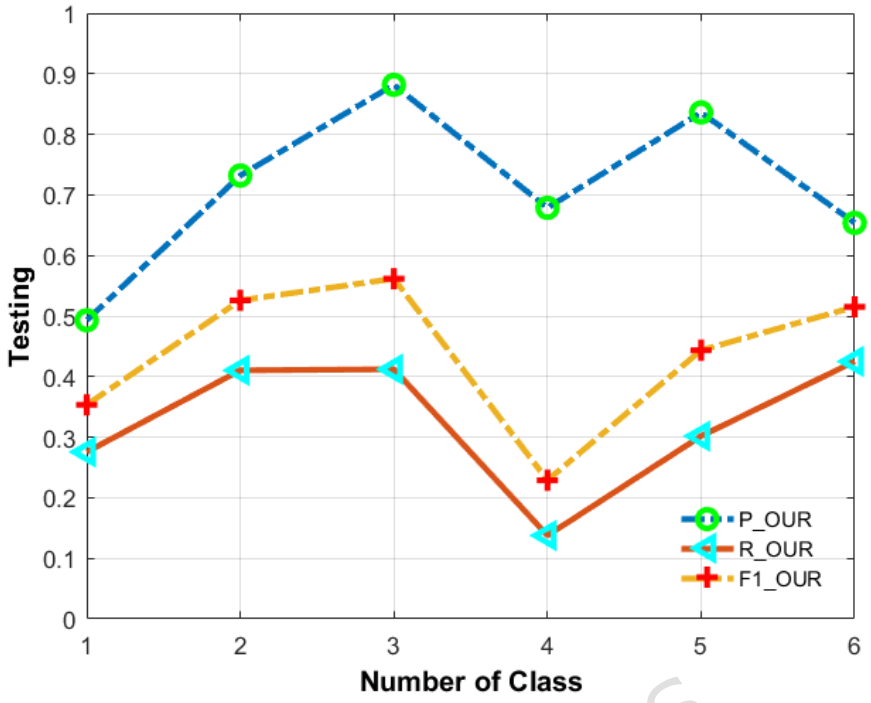
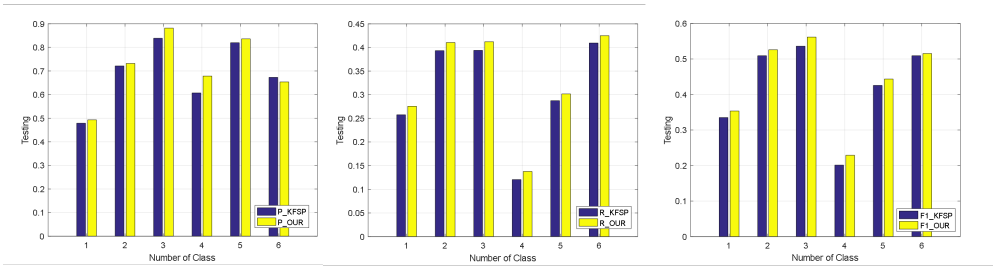


Fig. 7 Evaluation of AAPSP algorithm

R and F1 scores of the model are significantly improved after POL, and the average values of P, R and F1 after POL reach 71.25%, 32.71% and 43.81% respectively, which verifies the effectiveness of the dynamic sample screening strategy to enhance the classification performance of the model.

4.3.3 Comparison before and after POL

In this section, the P, R and F1 of the KFSP algorithm and the AAPSP algorithm proposed in this paper (i.e., the complete model of KFSP fused with POL) are compared on the ancient architecture dataset. The results are shown in Fig. 8. Experiments show that the indexes of the test set optimized by POL are significantly higher than those of the KFSP model before optimization. This result is due to the fact that POL iteratively filters misclassified samples with high fuzziness, which makes the model more targeted in the learning of key difficult samples, thus providing stronger generalization ability for the original misclassified samples in KFSP. Specifically, after the introduction of the active learning mechanism, the P value of the model is increased by 2.26% on average, the R value is increased by 1.67% on average, and the F1 value is increased by 1.88% on average, which further verifies the gain effect of the POL module on the KFSP performance.



(a) P (b) R (c) F1

Fig. 8 Comparison of KFSP and AAPSP

Table 2 Comparison of KFSP and UCSP. The table compares the training and testing accuracy of UCSP and KFSP in its base learner and meta-learner.

Methods	Models	Training accuracy	Testing accuracy
UCSP	Base learner1	82.38 ± 0.00	59.69 ± 0.02
	Base learner2	82.71 ± 0.01	58.70 ± 0.01
	Base learner3	81.93 ± 0.01	62.34 ± 0.02
	Meta Learner	89.72 ± 0.00	66.58 ± 0.01
KFSP	Base learner1	84.98 ± 0.00	61.12 ± 0.06
	Base learner2	85.39 ± 0.00	59.20 ± 0.01
	Base learner3	85.18 ± 0.00	59.42 ± 0.04
	Meta Learner	92.25 ± 0.00	$\mathbf{70.04 \pm 0.00}$

4.4 Accuracy comparison

4.4.1 Accuracy comparison before and after stacking

In this section, the average classification accuracy of KFSP and UCSP algorithms before and after stacking is compared on the ancient architecture dataset (expressed as Mean (%) ± STD (%)). The results are shown in Table 2 (where the black thickening term is the optimal result). The experimental data show that the overall performance of KFSP algorithm is better than that of UCSP algorithm. Specifically, After stacking, the average test accuracy of KFSP is 8.92%, 10.84% and 10.62% higher than that of its three base learners respectively, and 3.46% higher than that of UCSP after stacking, which further verifies the effectiveness of feature distribution constraint and AM to improve the performance of the model.

4.4.2 Accuracy comparison before and after POL

In this section, the average classification accuracy of the KFSP model before POL optimization and the AAPSP model formed after POL optimization (expressed as Mean (%) ± STD (%)) is compared on the ancient building data set. The results are shown in Table 3 (where the black bold item is the optimal accuracy). It can be seen from the table that after POL optimization, the average test accuracy of the AAPSP

Table 3 Comparison of KFSP, Our_all and AAPSP. The table gives the training and testing accuracy of KFSP, Our_all and AAPSP (Ours).

Model	Training accuracy (%)	Test accuracy (%)
KFSP	92.25 ± 0.00	70.04 ± 0.00
Our_all	86.92 ± 0.47	71.04 ± 0.89
AAPSP (Ours)	90.27 ± 0.01	**72.47± 0.35**

model is 2.43% higher than that of the KFSP, which fully verifies the effectiveness of the proposed POL model in enhancing the generalization performance of the model.

4.4.3 Comparison of AAPSP and Our_all algorithms

This section compares the average classification accuracy of the AAPSP algorithm and Our_all algorithm on the ancient architecture dataset (expressed as Mean (%) ± STD (%)), where Our_all algorithm refers to the results of KFSP training performed directly on all training images. The specific data is shown in Table 3. The results show that the average accuracy of AAPSP algorithm is significantly higher than that of Our_all algorithm.

In summary, under the premise of the same total amount of training images, the performance of the proposed algorithm (that is, by dividing the training image into an initial training set and a candidate set, and iteratively screening high-value samples from the candidate set to optimize the KFSP) is better than the training mode of one-time use of full training images, which verifies the advantages of the dynamic sample screening mechanism to improve the efficiency of the model.

5 Conclusion

In this paper, AAPSP is proposed to solve the problems of random projection feature weight mismatch and limited model generalization ability in ancient architecture image classification. This method uses the KFSP model to constrain the feature distribution of the weight matrix and weight the AM to achieve accurate focus on the key features of ancient buildings (such as bucket arch shape and roof contour), and combines the dynamic sample screening strategy of the POL model to effectively reduce the interference of redundant information and enhance the generalization ability of the model to cross-regional and multi-variant ancient buildings. The effectiveness of the proposed framework in the image classification task of ancient buildings is verified on six datasets of Chinese ancient buildings, which provides theoretical support and technical reference for image intelligent classification in digital protection of cultural heritage.

A key limitation of this work is that the model suffers from degraded classification performance on few-sample categories (e.g., the 4th category with limited training samples). Insufficient training data prevents the model from adequately learning the distinct feature patterns of such categories, leading to weak feature discriminability and a significant drop in classification accuracy. In future work, we will optimize the

model to better adapt to few-sample learning scenarios, strengthen feature representation and discriminability, and further improve classification accuracy for small-sample categories.

Declarations

The authors have no relevant financial or nonfinancial interests to disclose.

Data availability statement

The datasets used during the current study are available from the corresponding author on reasonable request.

Funding

This work was supported by the Scientific Research Funding Project for Outstanding Doctors (Post-doctors) Working in Shanxi Province, China (Grant No. 20025LJ020), the Taiyuan Institute of Technology Scientific Research Initial Funding, China (Grant No.2024KJ032), the Key Research and Development Plan of Shanxi Province (Grant No. 202202150401005), and the Special Research Project of Science and Technology Strategy in Shanxi Province (Grant No. 202404030401104).

Author contributions

All authors contributed to the current work. Conceptualization: C.Z., Z.S. Methodology: C.Z., S.X., Z.S. Formal analysis: C.Z., S.X., Z.Z., W.P. Example analysis: C.Z., S.X., W.P. Writing-original draft: C.Z. Writing-review: S.X., Z.S. Supervision: Z.S., Z.Z. All authors read and approved the final manuscript.

References

1. Wang.C et al (2025) Assessment of multiscale variability and influencing factors of humidity environment in cave temples of Wudang Mountain. Journal of Cultural Heritage 74:66-79

2. Jia.S et al (2022) Conservation and management of Chinese classical royal garden heritages based on 3D digitalization-A case study of Jianxin courtyard in Jingyi garden in fragrant hills. Journal of Cultural Heritage 58:102-111

3. Liao.B, Zuo.H, Yu.Y, Li.Y (2024) GraphMriNet: a few-shot brain tumor MRI image classification model based on Prewitt operator and graph isomorphic network. Complex & Intelligent Systems 10(5):6917-6930

4. Yang.B, Ding.L, Li.J, Li.Y et al (2025) Transformer-based multiple instance learning network with 2D positional encoding for histopathology image classification. Complex & Intelligent Systems 11(5):1-17

5. Ottoni.A.L.C, Ottoni.L.T.C et al (2025) A deep learning approach for cultural heritage building classification using transfer learning and data augmentation. Journal of Cultural Heritage 74:214-224

6. Ramirez-Arellano.A et al (2025). Deng entropy and LSTM neural network to classify built cultural heritage with severe damage. Journal of Cultural Heritage 73:286-294.

7. He.L, Wei.S, Zhou.H, Hu.Q (2025) Defending deep learning models: a hybrid algorithm employing de-noising and coordinate-disruption techniques against adversarial attacks. Complex & Intelligent Systems 11(10):1-17

8. Yang.J, Wang.L, Han.J, et al (2025) An air combat maneuver decision-making approach using coupled reward in deep reinforcement learning. Complex & Intelligent Systems 11(8):1-17

9. Artopoulos.G et al (2023) An artificial neural network framework for classifying the style of cypriot hybrid examples of built heritage in 3D. Journal of Cultural Heritage 63:135-147

10. Werbos.P et al (1981) Applications of advances in nonlinear sensitivity analysis. System Modeling and Optimization, Proceedings of the 10th IFIP Conference:3-12

11. Guo.P, Chen.C, Sun.Y et al (1995) An exact supervised learning for a three-Layer supervised neural network. Proceedings of the International Conference on Neural Information Processing ICONIP'95

12. Guo.P, Yu.M.L et al (2001) Pseudoinverse learning algorithm for feedforward neural networks. Advances in Neural Networks and Applications, Athens, Greece: World Scientific and Engineering Society Press 321-326

13. Mezina.A, Burget.R et al (2025) EnsArtNet: ensemble neural network architecture for identifying art styles from paintings. Journal of Cultural Heritage 72:71-80

14. Guo.P, Yu.M.L et al (2001) A case study on stacked generalization with software reliability growth modeling data. Neural Information Processing-18th International Conference ICONIP 2001

15. Guo.P, Yu.M.L et al (2004) A pseudoinverse learning algorithm for feedforward neural networks with stacked generalization applications to software reliability growth data. Neurocomputing 56(1):101-121

16. Li.S et al (2018) A hierarchical model with pseudoinverse learning algorithm optimazation for pulsar candidate selection. 2018 IEEE Congress on Evolutionary Computation (CEC)

17. Feng.S et al (2019) An ensemble model for error modeling with pseudoinverse learning algorithm. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)

18. Affentranger.F, Schneider.R et al (1992) Random projections of regular simplices. Discrete & Computational Geometry 7(3):219-226

19. Omidiran.D, Wainwright.M.J et al (2010) High-dimensional variable selection with sparse random projections: measurement sparsity and statistical efficiency. Journal of Machine Learning Research 11:2361-2386

20. Eftekhari.A, Babaie-Zadeh.M, Abrishami Moghaddam.H et al (2015) Two-dimensional random projection. Signal Processing 91(7):1589-1603

21. Young.J, Ridzal.D et al (2012) An application of random projection to parameter estimation in partial differential equations. Siam Journal on Scientific Computing 34(4):A2344-A2365

22. Nguyen.T.T et al (2019) Weighted multiple classifier framework based on random projection. Information Sciences 490:36-58

23. Lin.J, Cevher.V et al (2020) Convergences of regularized algorithms and stochastic gradient methods with random projections. Journal of Machine Learning Research 21(20):1-44

24. Zhang.B, Xiong.D.Y, Su.J et al (2020) Neural machine translation with deep attention. IEEE Transactions on Pattern Analysis and Machine Intelligence

25. Li.X, Yuan.A, Lu.X et al (2016) Vision-to-Language tasks based on attributes and attention mechanism. IEEE Transactions on Cybernetics, 99:1-14

26. Li.C et al (2020) ASIF-Net: attention steered interweave fusion network for RGB-D salient object detection. IEEE Transactions on Cybernetics 99:1-13

27. Zhao.Z, Zhang.W.Q et al (2021) End-to-end keyword search system based on attention mechanism and energy scorer for low resource languages. Neural Networks 139:326-334

28. Ahmad.M et al (2017) Fuzziness-based active learning framework to enhance hyperspectral image classification performance for discriminative and generative classifiers. Plos One 13(1):e0188996

29. Ahmad.M et al (2020) Spatial-prior generalized fuzziness extreme learning machine autoencoder-based active learning for hyperspectral image classification. Optik 206:163712

30. Pasolli.E et al (2014) SVM active learning approach for image classification using spatial information. IEEE Transactions on Geoscience & Remote Sensing

52(4):2217-2233

31. Li.J, Bioucas-Dias.J.M, Plaza.A et al (2015) Spectral-Spatial classification of hyperspectral data using loopy belief propagation and active learning. IEEE Transactions on Geoscience and Remote Sensing 51(2):844-856

32. Shi.Q, Du.B, Zhang.L et al (2015) Spatial coherence-based batch-mode active learning for remote sensing image classification. IEEE Transactions on Image Processing 24(7):2037-2050

33. Li.J, Bioucas-Dias.J.M, Plaza.A et al (2011) Hyperspectral image segmentation using a new bayesian approach with active learning. IEEE Transactions on Geoscience and Remote Sensing 49(10):3947-3960

34. Lewis.D.D et al (1994) A sequential algorithm for training text classifiers. Springer London:3-12

35. Di.W, Crawford.M.M et al (2011) Active learning via multi-View and local proximity co-regularization for hyperspectral image classification. IEEE Journal of Selected Topics in Signal Processing 5(3):618-628

36. Hinke et al (1991) Query by committee. University of Southern California

37. Rajan.S, Ghosh.J, Crawford.M.M et al (2008) An active learning approach to hyperspectral data classification. IEEE Transactions on Geoence and Remote Sensing

38. Ahmad.M et al (2019) Patial prior fuzziness pool-based interactive classification of hyperspectral images. Remote Sensing 11(9):1136

39. Pasolli.E et al (2011) Improving active learning methods using spatial information. Geoscience & Remote Sensing Symposium

40. Liu.A, Jun.G, Ghosh.J et al (2009) Active learning of hyperspectral data with spatially dependent label acquisition costs. Geoscience & Remote Sensing Symposium

41. Tuia et al (2011) A survey of active learning algorithms for supervised remote sensing image classification. IEEE Journal of Selected Topics in Signal Processing, 5(3):606-617

42. Haines.T.S.F et al (2011) Active learning using dirichlet processes for rare class discovery and classification. Ieice Technical Report Microwaves

43. Michel, Malik, Inglada et al (2010) Lazy yet efficient land-cover map generation for HR optical images. 2010 IEEE International Geoscience and Remote Sensing Symposium

44. Demir, Persello, Bruzzone et al (2011) Batch-mode active-learning methods for the interactive classification of remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 49(3):1014-1031

45. Munoz-Mari.J, Tuia.D, Camps-Valls.G et al (2012) Semisupervised classification of remote sensing images with active queries. IEEE Transactions on Geoscience & Remote Sensing 50(10):3751-3763

46. He.L et al (2018) Recent advances on spectral-spatial hyperspectral image classification: an overview and new guidelines. IEEE Transactions on Geoscience and Remote Sensing 56(3):1579-1597

47. (2018) Datasets for 3D reconstruction. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences [Online]. Available: http://vision.ia.ac.cn/data

48. (2024) Wei.G Chinese Ancient Architecture Image Database. V1. Science Data Bank. https://doi.org/10.57760/sciencedb.ai.00008