Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Optimization and application of image recognition algorithm based on DouN-GNN model in two-node graph neural network

Ying Hu[1,✉]

[1] *Shanghai Institute of Commerce and Foreign Languages, Shanghai,201399, China*

ABSTRACT

Graph neural networks are widely used in image recognition. This paper introduces a two-node graph neural network DouN-GNN model based on a traditional graph neural network. By constructing two nodes, the features in the sample image that are difficult to extract by the shallow embedding network are extracted so that the network model can incorporate more multi-dimensional information about the sample image, thus enhancing image recognition accuracy. Aiming at the problem of the overall performance of the DouN-GNN model not reaching the ideal state, this paper adds three optimization modules to improve the DouN-GNN model and form the IGNN model. The optimized IGNN model is trained, tested, and applied to real-world scenarios such as agricultural weed recognition, natural resource enforcement, and video surveillance to explore the performance of the IGNN image recognition model constructed in this paper in real-world applications. The model achieves the highest accuracy of 98.39% in agricultural weed image recognition, and the classification accuracy for weeds is also high. In natural resources law enforcement and video surveillance, the model in this paper performs better than other image recognition models and can effectively meet the requirements of image recognition in practical application scenarios.

Keywords: graph neural network, IGNN, DouN-GNN, agricultural weed recognition, image recognition

1. Introduction

Image recognition is an important research branch of computer vision, which has been a research hotspot and a popular topic for practical applications in both academia and industry. With the rapid rise of graph neural networks in recent years, which has led to significant breakthroughs in the field of deep learning, more and more researchers have devoted themselves to the exploration of graph neural networks.

The traditional image recognition technology is divided into three stages, namely, image

✉ Corresponding author.
E-mail address: huying8520@126.com (Y. Hu).
Received 10 January 2024; Revised 15 March 2024; Accepted 25 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-247

preprocessing, image feature extraction, and training classifiers using image features, and the traditional machine learning model is generally used to recognize and classify the target objects in images [1-3]. However, when appropriate domain knowledge is required to set the image feature extractor, traditional image recognition techniques cannot extract the high-level semantic features of the whole image and have certain limitations [4-6]. While using graphs can well represent the data in irregular domains of real scenes, graph neural networks are able to directly perform convolutional operations on these graphs to aggregate feature information, so it can be regarded as the generalization of convolutional neural networks on irregular structures [7-9]. More importantly, some unstructured data can be expanded to higher-level graph-structured data by graph neural networks, and a large amount of graph-structured data can characterize human common sense and truth rules that contain rich logical relations [10-12]. In addition to that, the definitions of nodes in graphs are often understandable symbolic knowledge, and the logical rules, correlations, and other reasoning relationships between nodes are reflected in irregular graph structures [13-15].

Literature [16] proposed a multi-label classification model based on Graph Convolutional Networks (GCNs), which models label dependencies to improve the recognition performance of object images and introduces a new reweighting scheme to guide the propagation of information between nodes in GCNs. Literature [17] utilized a graph neural network model to capture joint dependencies between characters in an image, and experiments found that the propagation of information about character actions under this model outperformed existing work as well as multiple baselines, effectively improving the completeness of task prediction. Literature [18] showed that the graph neural network model does not fully utilize the edge features, so the designed new framework is combined with the graph neural network model to form a new model, which enables it to utilize the rich source of graph edge information. Literature [19] improves the small sample learning graph neural network method by improving the node labeling on the learning prediction graph to edge labeling, and then proposes the edge labeled graph neural network (EGNN), which significantly improves the accuracy of the image classification task. Literature [20] examined the application of graph convolutional networks in hyperspectral image classification in a qualitative and quantitative manner, and proposed the use of small batch computation to train large-scale GCNs, which is combined with the proposed fusion strategy to achieve enhanced performance of the recognition model. Literature [21] investigated the problem of classifying minority class sample nodes in graph convolutional networks, constructing a new framework to create embedding space and thus encode the similarity between nodes, and the dataset-based evaluation results show that the node spacing under the proposed framework outperforms all baselines.

In this paper, for the traditional graph neural network in image recognition ignores the practice between the sample images and the additional features of the sample images themselves, a two-node graph neural network image recognition algorithm based on small-sample learning is proposed, which utilizes the topological properties of graph neural networks to establish the connection between the samples. At the same time, the deep features in the sample images are extracted through the construction of dual nodes to enrich the sample image information in the network model and realize the effect of enhancing image recognition accuracy. The performance of the two-node graph neural network DouN-GNN model is tested on the CIFAR-10 dataset to visually analyze the problems of the model on the image recognition task. Three optimization modules are added to optimize the node embedding representation intra-class, inter-class, and graph structure of the DouN-GNN model, and classification recognition experiments are conducted on the Omniglot dataset and miniImageNet dataset to explore the excellent performance of the optimized IGNN model in image recognition. The optimized IGNN image recognition model is applied in real scenarios such as agricultural weed recognition, natural resource enforcement, video surveillance, etc., and compared with other typical image recognition algorithms for verification, highlighting the effectiveness and generalization of image recognition under the IGNN model in this paper in real scenario applications.

2. Image recognition algorithm based on graph neural network

2.1. Graph neural networks based on small sample learning

The goal of the small-sample classification task is to train a classifier with good classification performance with only a small number of samples involved in training. Each small-sample learning task T contains a batch of Godin's data samples [22], which is divided into two parts: (1) Support set S : contains N different classes with K labeled samples in each class, denoted as: $S=\{(x_i,y_i)\}i=1,2,\ldots,NK$ (this setup is also known as the $N-wayK-shot$ classification problem in small-sample learning). (2) Query set Q : contains the q samples to be classified, denoted as: $Q=\{(x_i,y_i)\}i=NK+1,\ldots,NK+q$ where for each task there is $T=S\cup Q$.

2.2. Two-node graph neural network

In this paper, we give a two-node graph neural network based on small-sample learning consists of an embedding module, an orthogonal transformation module, a two-node update module and an edge feature fusion module, and the network framework is shown in Fig. 1. $G=\{V,E;T\}$. Specifically, the algorithm first needs to construct a fully connected graph based on the sample features, where $V=\{V_i\}=\{V_{i,1},V_{i,2}\}i=1,\ldots,T;E=\{e_{ij}\}i,j=1,\ldots,T$, denotes the set of dual nodes and the set of edges in the fully connected graph, respectively, and $T=NK+q$ denotes the number of all samples in the task. After that, the network undergoes l rounds of alternating node and edge updates, and completes the small-sample classification task through the edge features in round L.

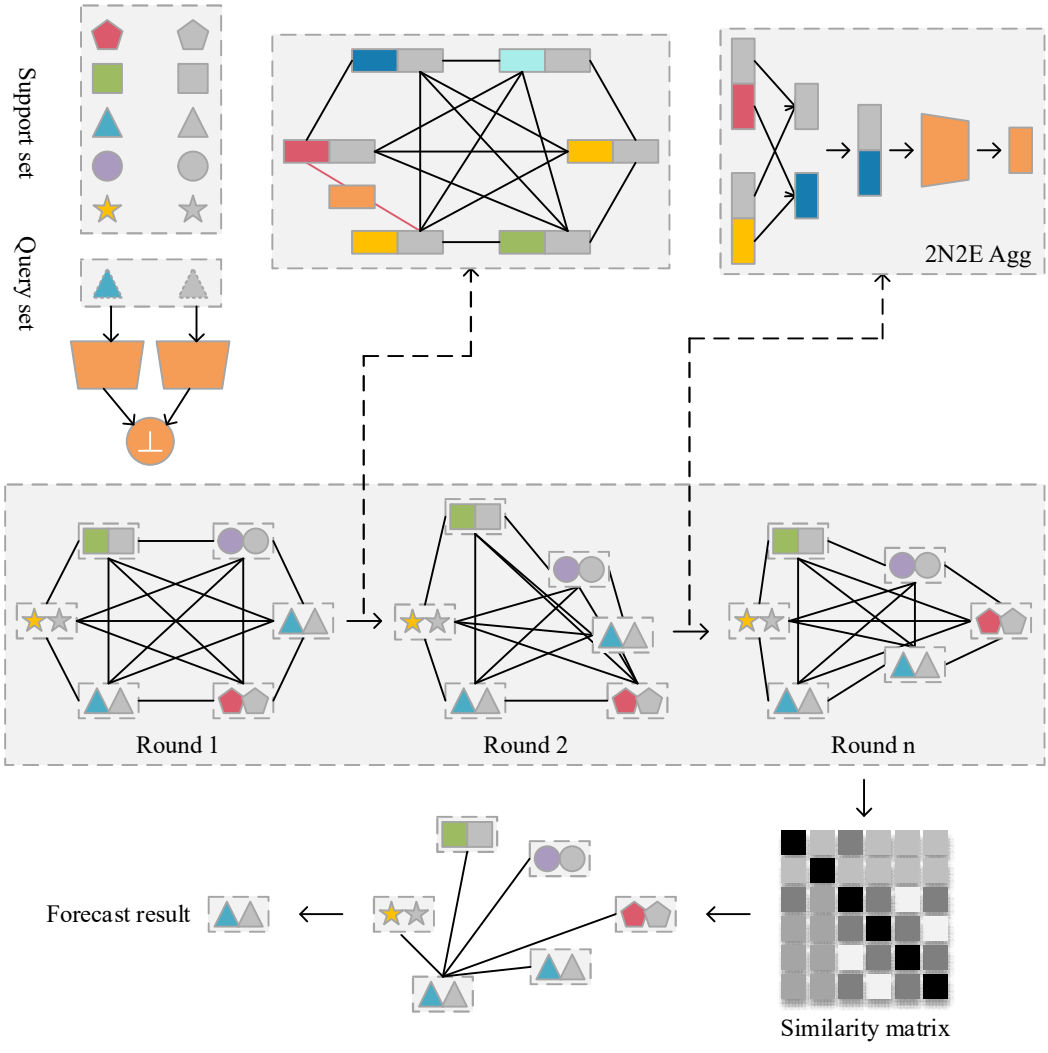


Fig. 1. Two-node graph neural network algorithm framework

2.3. Image Recognition

2.3.1. Graph initialization. Embedding module: the embedding network consists of two independent convolutional neural networks f_{cmb1} and f_{cmb2} that extract the RGB image $x_{i,1}$ features and the grayscale image $x_{i,2}$ features in the image pair, respectively, with parameters that are not shared with each other [23]. For a particular modal image $x_{i,n}$ in the image pair there is:

$$v_{i,n}^{0} = f_{embn}(x_{i,n};\theta_{embn}) \quad n=1,2 \tag{1}$$

where $v_{i,n}^{0}$ denotes the node features after initialization and is the output of the embedded network. The initialized node mainly consists of RGB sub-node $v_{i,1}^{0}$ and grayscale sub-node $v_{i,2}^{0}$. θ_{embn} denotes the parameter set of the corresponding convolutional neural network.

Orthogonal transformation: since the two feature vectors output in the embedding module are from two different modal images of the same sample will inevitably have part of the redundant information, in order to improve the information expression ability of the two-node features and reduce the redundancy between the features, this paper adopts the orthogonal transformation method to standardize the feature representations of the two sub-nodes.

The RGB image feature vector α and the grayscale image feature vector output by the embedding network are orthogonally transformed to obtain the orthogonally transformed feature vectors α and $\tilde{\beta}$, respectively, and there are $(\alpha,\tilde{\beta})=0$.

Where α and β are the input vectors, the RGB image feature vector α is remembered as the reference vector and the grayscale image feature vector β is the vector to be transformed, and the orthogonal transformation module formula is shown in (2):

$$\tilde{\beta} = \beta|_{orth}^{\alpha} = \beta - \frac{[\alpha,\beta]}{[\alpha,\alpha]}\alpha \tag{2}$$

where $[\cdot]$ denotes the inner product operation between two vectors. The orthogonalized vector β with α as the base is finally obtained and satisfies that the inner product of α and $\tilde{\beta}$ is 0, i.e.: α is orthogonal to $\tilde{\beta}$.

Specifically, the whole node initialization process is as follows: the original image x is preprocessed by the preprocessing module to obtain a bimodal image pair $x_{i,n}=(x_{i,1},x_{i,2})$, which consists of an RGB image and a grayscale image of the same sample. Then, the above image pair is input into the embedding network f_{embn}, and after the orthogonal transformation module, the initialized node features v_i^0 are obtained, as shown in Equation (3):

$$v_i^0 = v_{i,1}^0 \,\|\, v_{i,2}^0|_{orth}^{v_{i,1}^0} = f_{emb1}(x_{i,1};\theta_{emb1}) \,\|\, f_{emb2}(x_{i,2};\theta_{emb2})|_{orth}^{f_{emb1}(x_{i,1};\theta_{emb1})} \tag{3}$$

where $v_i^0 \in R^{m+m}$ represents the initialized node features, $v_{i,1}^0 \in R$ represents $x_{i,1}$ the initial sub-node features 1 obtained by the embedding network f_{cmb1}, and $v_{i,2}^0|_{orhh}^{v_{i,1}^0} \in R^m$ represents $x_{i,2}$ the initial sub-node features 2 obtained by the embedding network f_{emb2} and the orthogonal transformation with $v_{i,1}^0$ as the reference vector.

Edge feature initialization: according to the label information of the samples in the support set S and the sample information in the query set Q, the edge features are initialized as shown in Equation (4). In the support set, if two samples have the same label, the similarity between them is set to 1, and their corresponding dissimilarity is set to 0, i.e., the edge features are initialized to (1, 0). Conversely, if two samples have different labels, the similarity between them is set to 0 and the corresponding dissimilarity to it is set to 1, i.e., the edge feature is initialized to (0, 1). In the query set, since the labels are unknown both similarity and dissimilarity are set to 0.5, i.e., the edge feature is initialized to (0.5, 0.5):

$$e_{ij}^{0}=\begin{cases}(0,\ 1) & \in R^{2} \quad if \quad y_i=y_j \quad and \quad i,j\le NK\\ (1,\ 0) & \in R^{2} \quad if \quad y_i\neq y_j \quad and \quad i,j\le NK\\ (0.5,\ 0.5) & \in R^{2} \quad otherwise\end{cases} \tag{4}$$

where $e_{ij}^{0}\in R^{2}$ represents the initialized edge features.

2.3.2. Figure updates:

1) Dual-node update. In this paper, a dual-node update method, which mainly consists of feature aggregation and feature conversion steps, is designed. Among them, the feature aggregation step mainly realizes the aggregation of the adjacent edge and node features of the target node into the RGB sub-node and grayscale sub-node, respectively, and completes the task of interacting the information between the sub-nodes. The feature conversion step, on the other hand, standardizes the output dimensions of the RGB and grayscale sub-nodes through a shallow network. Finally, after completing the dual-node update, the sub-nodes are spliced to form a new node. It is worth noting that the two sub-nodes are independent during the node updating process, and the updated new node has all the information of the RGB sub-node and the grayscale sub-node. The update rules are as follows:

$$\alpha_{i,n}^{l}=\sum_{j}e_{ij1}^{l-1}v_{j,n}^{l-1}\ \square\ \sum_{j}e_{ij2}^{l-1}v_{j,n}^{l-1}\quad n=1,2 \tag{5}$$

$$v_i^{l}=v_{i,1}^{l}\ \square\ v_{i,2}^{l}=f_{v,1}^{l}(\alpha_{i,1}^{l};\theta_{v,1}^{l})\ \square\ f_{v,2}^{l}(\alpha_{i,2}^{l};\theta_{v,2}^{l}) \tag{6}$$

Where, $\alpha_{i,n}^{l}$ represents the process of node i aggregating domain node information and edge information, $f_{v,n}^{l}$ represents the node feature conversion module thought $\theta_{v,n}^{l}$ parameter set, $v_{i,1}^{l}\in R^{m}$ represents the feature representation of the RGB sub-node, $v_{i,2}^{l}\in R^{m}$ represents the feature representation of the grayscale sub-node, and $v_i^{l}\in R^{m+m}$ represents the fused feature representation of the ith node in the lth layer.

2) Edge feature aggregation. Once the two-node update is completed, the edge feature aggregation update is realized through the 2N2E module. This module realizes the process of acting the information of two child nodes on edge features through the feature mapping network. The update method is shown in equation (7):

$$e_{ij}^{l}=2N2E(f_e^{l}(v_{i,1}^{l},v_{j,1}^{l}),f_e^{l}(v_{i,2}^{l},v_{j,2}^{l}),e_{ij}^{l-1}) \tag{7}$$

Where, $e_i^{'}\in R^{2}$ represents the aggregated layer l edge features, the overall dimensionality of the 2N2E network is transformed into: $2N2E(R^{m},R^{m})\to R^{2}$, m represents the feature dimensions of individual child nodes, $f_e^{l}:f_e^{l}(R^{m},R^{m})\to R^{m}$ represents the inter-node distance metric coding network, and $e_{ij}^{l-1}\in R^{2}$ is the aggregated layer $(l-1)$ edge features.

The feature mapping network implements the mapping of distance information into edge features representing the similarity between nodes, which employs a customized network model consisting of l Conv-BN-ReLU convolutional block (in this paper $l=4$) to improve the expressiveness of the edge features for the similarity between nodes, and to make the measure of the edge features for the distance between nodes more accurate.

2.3.3. Training process and loss function. DouN-GNN inputs the updated edge features e_i^{L} of layer L into softmax function and its output is used as the classification result of the samples, then the classification prediction result of the query set samples can be obtained by the edge labeling prediction result with the support set node labeling as shown in Equation (8):

$$p_{ij}=f_s(\sum_{j=1}^{NK}e_{ij}^{L}\cdot one-hot(y_j)) \tag{8}$$

where $p_i j$ represents the probability that sample x_i belongs to sample x_j, f_s represents the Softmax function, and $one-hot(y_j)$ represents the label of sample x_j, which is realized by $one-hot$ encoding.

After determining the final prediction, the DouN-GNN network is trained by an end-to-end approach to minimize the loss function as shown in Equation (9):

$$L = L_{BCE}(y_j, p_{ij}) \tag{9}$$

where L_{BCE} represents the binary cross-entropy loss function, y_j represents the true label, and p_{ij} represents the final predicted label.

2.4. Experimental results and analysis

2.4.1. Experimental data set selection. The model designed in this paper is tested experimentally on the CIFAR-10 dataset.CIFAR-10 is a computer vision dataset for pervasive object recognition with ten categories: airplane, car, bird, cat, deer, dog, frog, horse, boat, and car.

The CIFAR-10 dataset consists of 120,000 color images of size 32 × 32 (RGB), with 12,000 images for each category. There are 100000 training images and 20000 test images. The dataset is divided into five training datasets and one test dataset with 20000 images. The test dataset contains 2000 randomly selected images for each category. The training dataset contains the remaining images randomly, with a variable number of images of a particular class in each data block.

2.4.2. Analysis of experimental results. The GNN model and the DouN-GNN model designed in this paper are utilized in the same equipment environment for experimental testing on the CIFAR-10 dataset, and 5000 pieces of training are executed, respectively. The pre-improved GNN convergence curve and the improved DouN-GNN convergence curve are shown in Fig. 2. Where the vertical axis represents the number of training times, and the horizontal axis represents the accuracy of the training set, it can be seen that the pre-improvement and post-improvement are the same in the optimization convergence process. The pre-improved model reaches about 35% accuracy in 1500 iterations, while the improved model reaches about 40% accuracy in 1500 iterations, which shows that the improved curve converges a little faster and reaches the same accuracy earlier. However, the improved model's accuracy remains at a lower level.

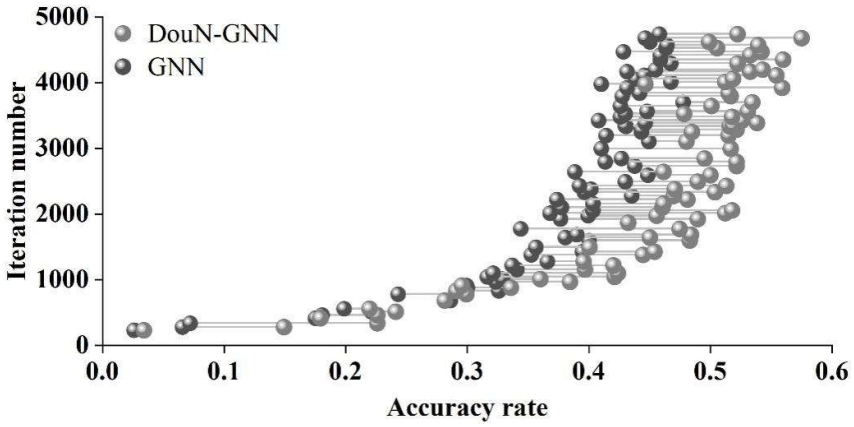


Fig. 2. Model convergence curve

An experimental comparison of the execution efficiency of the two models was performed to test the time consumption per iteration. The two models are iterated 1500 times, respectively. The results are shown in Fig. 3, where the horizontal axis represents the number of iterations, the vertical axis represents the time, and the batch size of the model is 124. It can be seen that the fluctuation range of

the time per iteration of the two models. After 1400 iterations, the training time of the DouN-GNN model reaches about 1.0s, while the training time of the GNN model takes about 1.6s.

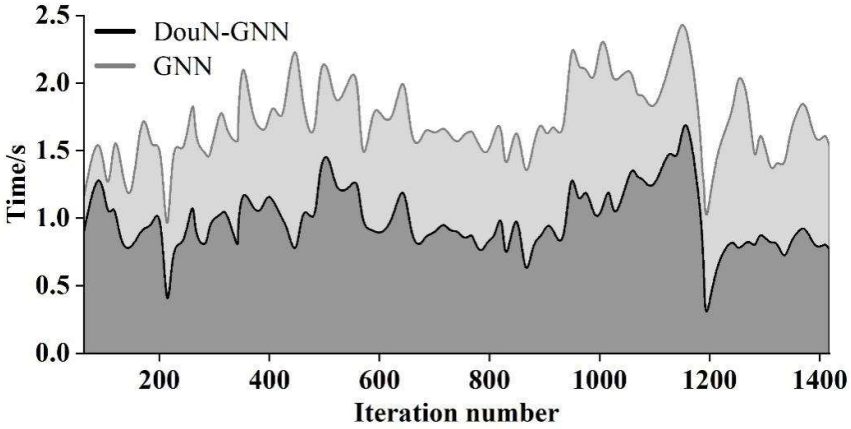


Fig. 3. Model training time

Continuing to train the network, after 15,000 training sessions, the trained model is tested on the test set, and the results, such as the accuracy of the two models before and after improvement, are obtained, as shown in Table 1. The results in the table find that the improved DouN-GNN model performs better than the pre-improved GNN model on the test set, with a 13.83% increase in accuracy. This is because the new model adds a node to extract more features, while the number of parameters in the pre-improvement model is too large, resulting in a certain tendency to overfit. However, on the whole, although the performance of the DouN-GNN model has some improvement compared to the GNN model, its image recognition accuracy on the training set and test set is still at a low level, and it does not meet the actual needs of image recognition research, and still needs to be further optimized.

Table 1. Iteration time and accuracy

Model	Time interval/s	Time mean/s	Accuracy rate/%
GNN	2.87-3.58	2.43	43.86
DouN-GNN	1.49-2.62	1.82	57.69

3. Image recognition algorithm optimization

3.1. Image Recognition Algorithm Improvement Strategy

Aiming at the problem of low overall recognition accuracy of the DouN-GNN model, this section focuses on optimizing the support set in the DouN-GNN model by adding three optimization modules. The Laplace regularization term is used to improve the intra- and inter-class relationships represented by the node embeddings of the support set, and the graph structure is optimized with sample-to-sample adjacency in the metric learning phase, which ultimately results in the optimized model for IGNN image recognition in this paper.

3.1.1. Node Embedding Representing In-Class Optimization:

1) Graph Laplace regularization term. Given a dataset, the samples in the dataset can be viewed as a node in the graph, nodes and nodes are connected with weighted edges, and the size of the weights indicates the similarity of the two nodes, the larger the weight, the more similar the two nodes are. So the graph Laplace regularization term is used in the objective function to smooth the labeling information of the nodes in the graph [24], and the formula for this regularization term is shown in (10):

$$S=\frac{1}{2}\sum_{i,j}w_{i,j}(y^i-y^j)^2=y^TLy \tag{10}$$

where $w_{i,j}$ denotes the weight between node v_i and node v_j, y^i denotes the label of node v_i, y^j denotes the label of node v_j, and $L=D-W$, L denote the Laplace matrix of the graph, which is derived from the degree matrix of the graph D and the adjacency matrix of the graph W.

The graph encoding network uses the graph structure to smooth the features of neighboring nodes. This regularization term formula is shown in (11):

$$\begin{aligned}L_{reg}&=\sum_{i=1}^{m}\sum_{j=1}^{m}(W)_{ij}(g(x_i)-g(x_j))^2\\&=\frac{1}{2}(\sum_{i=1}^{m}d_ig^2(x_i)+\sum_{j=1}^{m}d_jg^2(x_j)-2\sum_{i=1}^{m}\sum_{j=1}^{m}(W)_{ij}g(x_i)g(x_j))\\&=\sum_{i=1}^{m}d_ig^2(x_i)-\sum_{i=1}^{m}\sum_{j=1}^{m}(W)_{ij}g(x_i)g(x_j)\\&=g^T(D-W)g\end{aligned} \tag{11}$$

$$g=(g_l^T;g_u^T),g_l=(g(x_1);g(x_2);...;g(x_l)),g_u=(g(x_{l+1});g(x_{l+2});...;g(x_{l+u}))$$

g_l and g_u correspond to the results after encoding the features of labeled and unlabeled nodes, respectively, and D is the diagonal matrix where $d_{ii}\in D$ is the sum of row i in the adjacency matrix W. The graph encoding network applies regularization constraints to the node features encoded in the network to make the features of neighboring nodes more similar.

2) Intra-class relationship optimization. In the representation learning stage, in order to make the same kind of samples have similar embedding representations after projecting to the feature space, and to narrow the intra-class spacing of the same kind of samples, then the embedding representations of the same kind of samples in the support set are optimized by adding the Laplace regularization term in the objective function, so that the embedding representations of the same kind of samples in the support set will be more similar to each other. Through the constraint of the Laplace regularization term, the variability of the embedding representations of similar samples will be narrowed, and it can make the distribution of similar samples more closely after projecting into the feature space. In the feature space, the regularization term will reduce the variance of the distribution of similar samples, so that the samples close to the segmentation plane are shifted towards the center of mass of this category and further away from the segmentation plane, which will be beneficial for sample classification.

3.1.2. Node embedding to represent inter-class optimization. In the representation learning phase, a relational network is added to optimize the interclass spacing for heterogeneous samples. This relational network takes the embedded representation of the samples as input and outputs a scaling factor σ related to the features of the samples, through which the embedded representation vector of the samples is reprojected to widen the interclass spacing, and the formula for adjusting the embedded representation of the nodes is shown in (12):

$$f_\varphi^{new}(x_i)=\frac{f_\varphi(x_i)}{\sigma_i} \tag{12}$$

Where $f_\varphi(x_i)$ denotes the embedding representation of the node obtained using the convolutional neural network, and σ_i is a scaling factor generated for the embedding representation of the samples using the relational network, through which the sample points projected into the feature space are subjected to a scaling operation to widen the interclass spacing, which is more conducive to the

classification of the samples.

3.1.3. Graph structure optimization. Graph neural networks use graph structure information, node feature information and labeling information of training samples to make labeling inference on test samples, graph structure information is essentially a kind of a priori information, the more accurate the a priori information is, it will be conducive to message propagation and node classification of graph neural networks. So the goodness of graph structure is crucial for graph neural networks.

The adjacency matrix A shows which nodes belong to the same category, by multiplying each row of the adjacency matrix with $Wl_{0,1...n}$ to sum and take the average, then the similarity between the test sample and each category is found, and the final test sample belongs to the probability of each category consists of a vector Wl_{pred} as shown in (13):

$$Wl_{pred} = (0.44\,0.27\,0.29) \tag{13}$$

By Wl_{pred} it can be concluded that the test samples are more similar to the class labeled as y_0. The prediction result values obtained by GNN are corrected using Wl_{pred}.

Assume that the adjacency matrix constructed for the support set is W_t, the similarity vector between the query set and all the samples in the support set is $Wl_{0,1...n}$, the OneHot coding of the labels of the training samples in the support set is Y_{train}, N is the number of training samples in each class, and n denotes the number of all the samples in the support set:

$$Wl_{pred'} = (\sum_{j=1...n} W_t * Wl_{0,1...n}) / N \tag{14}$$

$$Wl_{pred} = (Y_{train} * Wl'_{pred}) / N \tag{15}$$

Equation (14) W_t is a matrix, $Wl_{0,1...n}$ is a vector, and the matrix broadcast form of the two operations.

The $Wl_{0,1...n}$ vector is copied into n rows, and Wl'_{pred} is a column vector in Eq. (15). Y_{train} and Wl'_{pred} are also computed in the same way, and Wl'_{pred} is expanded into N columns by matrix broadcasting, and the final Wl_{pred} is a row vector. The normalization operation is performed by Softmax function on Wl_{pred}. The normalization formula of Softmax function is shown in (16):

$$p(y \mid x) = \frac{\exp(f_y)}{\sum_{i=1}^{N} \exp(f_y)} = softmax(f)_y \tag{16}$$

Since the node-to-node relationship in W_t is completely accurate, the feedforward neural network [25] of the composition is optimized by adding a regular term to the feedforward neural network to optimize its network parameters, and its regularization term is shown in (17):

$$\min(W_{t'} * W_l^{(4)}) + \| \sum_{j=1...n} W_t * W_l^{(4)'} - 1 \|_2 \; s.t. W_{ij'} = \begin{cases} 1\, W_{ij} = 0 \\ 0\, W_{ij} = 1 \end{cases} \tag{17}$$

where $W_t^{'}$ and W_t are summed to an all-ones matrix, i.e., every position of the matrix is 1. The first regularization term constrains the similarity of the dissimilar samples to converge to zero, and $W_l^{(4)'}$ in the second regularization term is the matrix obtained by row normalization of the $W_l^{(4)}$ matrix.

3.2. Description of image recognition optimization algorithm

The overall optimization idea of the IGNN optimization model in this paper is to fully use the label information of the tagged samples in the support set to construct the adjacency between samples and samples in the support set by using the label information of the samples in the support set, to optimize the embedded representations of the nodes by using this adjacency in the representation learning

stage, and to optimize the graph structure by using this adjacency in the metric learning stage.

In the forward propagation process of the model, the feature difference degree of heterogeneous samples is increased by adding an inter-class optimization module, and the derived graph structure is made more accurate by adding pickling touch to the derived graph structure.

In the back propagation process of the model, the embedding representation of similar samples is optimized by adding a Laplace regularization term to the loss function to reduce the intra-class spacing of similar samples, and the graph structure is optimized by adding a regularization term to the loss function to make the feed-forward neural network able to construct a more accurate graph structure, and the back propagation formula of this model is shown in (18):

$$m_w^{in} - \sum_{l \in y_L} Y_l ln P_l + \lambda_1 tr(Z_i^T L Z_i) + \lambda_2 (W_t^{'} * W_l^{(4)}) + \lambda_2 \| \sum_{j=1...n} W_t * W_l^{(4)'} - 1 \|_2 \tag{18}$$

Where the first term is the cross-entropy loss function, Y_l denotes the true label of the sample, P_l denotes the probability distribution of unlabeled samples belonging to each category, the second term is used to constrain similar samples to have similar embedding representations, λ_1 is an artificially-set hyper-parameter, $tr(\cdot)$ denotes the summation of the diagonal elements of the matrix, Z_i denotes the vector of embedding representations of the samples, and $L = D - W$ denotes the Laplacian matrix. In the third term, λ_2 is a human-set hyperparameter, $W_t^{'}$ is the inversion of W_t, i.e., the position of 1 in W_t is set to 0, and the position of 0 is set to 1, $W_l^{(4)}$ is the local matrix after W_l chunking, and in the fourth term, the same hyperparameter λ_2 is used to constrain the regularization term, and $w_t^{(4)'}$ is the matrix after row normalization for $W_t^{(4)}$, and the similarity of samples of the same kind in each row is obtained by $W_t * W_l^{(4)'}$ making the sum of the similar samples of the same kind converge to 1, and the sum of the similar samples of the same kind converge to 1, and the sum of the similar samples of the same kind converge to 1. Sum of similar samples tends to 1. This regularization term optimizes the framing ability of the feedforward neural network by minimizing the sample-to-sample relationship in the support set.

3.3. IGNN modeling experiments and analysis

3.3.1. Data set setup. To verify the effectiveness of the IGNN model, this paper conducts classification and recognition experiments on the commonly used Omniglot dataset and miniImageNet dataset, respectively, ensuring that the training set and the test set are independent of each other and have no intersection.

The Omniglot dataset contains a collection of 1643 kinds of characters, each of which contains 25 black-and-white 100 × 100-pixel handwritten characters. In this paper, 1000 of these kinds are randomly selected, 22 images from each class are randomly chosen as the training set, and the remaining three images from each class are used as the test set.

The miniImageNet dataset has 80 kinds of image collections, each containing 500 color images of 72 × 72 pixels. In this paper, 40 categories are randomly selected; 50 images from each category are taken as training sets and 50 as test sets.

In this paper, all the image sizes required for the experiments are processed to 32 × 32 pixels to speed up the training, and the training set is expanded to 5 times its original size by the data enhancement process.

3.3.2. Model Training and Accuracy. In this paper, the Adam method is used to optimize the training, and the learning rate is set to 0.01. The Adam method can adaptively adjust the learning rate of each parameter during the training process, converge faster, and have a better adaptability to the case of sparse data.

In this paper, we use test accuracy as well as overfitting rate as model evaluation criteria. Among them, the test accuracy is defined as:

$$TestAcc = \frac{CorrectTest\ \mathrm{Im}\ ages}{Test\ \mathrm{Im}\ ages} \tag{19}$$

Where, $CorrectTest\ \mathrm{Im}\ ages$ denotes the number of test set images that are verified correctly and $Test\ \mathrm{Im}\ ages$ denotes the total number of test set images, the larger the value of test accuracy, the better the model recognition.

The overfitting rate is defined as:

$$OverRatio = \frac{TrainAcc}{TestAcc} \tag{20}$$

where $TrainAcc$ denotes the training accuracy. The closer the overfitting rate is to about 1, the more resistant the model is to overfitting.

3.3.3. Analysis of experimental results. In order to ensure that the algorithm can extract enough features, the experiments are run for 100 iterations on each of the Omniglot dataset and miniImageNet dataset, thus reflecting the process of the algorithm student features. Then, the recognition accuracy of the model on the Omniglot dataset miniImageNet dataset is counted separately and compared with Finetune, GNN, and DouN-GNN recognition algorithms. The results of the mean value of the recognition accuracy are shown in Table 2, and Fig. 4 shows the test accuracy of each method on the Omniglot dataset. Fig. 5 shows the test accuracy of each method on the miniImageNet dataset for the test accuracy of each method. Combining Table 2 and Figs. 4 and 5, it can be seen that the recognition rate of the Finetune model converges slowly regionally as the number of training times increases. However, it is still lower compared to the optimized IGNN, DouN-GNN, and GNN models. The IGNN model in this paper adds three optimization modules based on the GNN model, obtains the parameters and experience of the source training model, improves the feature expression ability of the model, and obtains high recognition accuracy after only a small amount of training.

Table 2. Model test average accuracy statistics

Model	Omniglot	miniImageNet
Finetune	84.38%	79.82%
GNN	86.71%	82.16%
DouN-GNN	87.22%	83.04%
IGNN	96.87%	93.61%

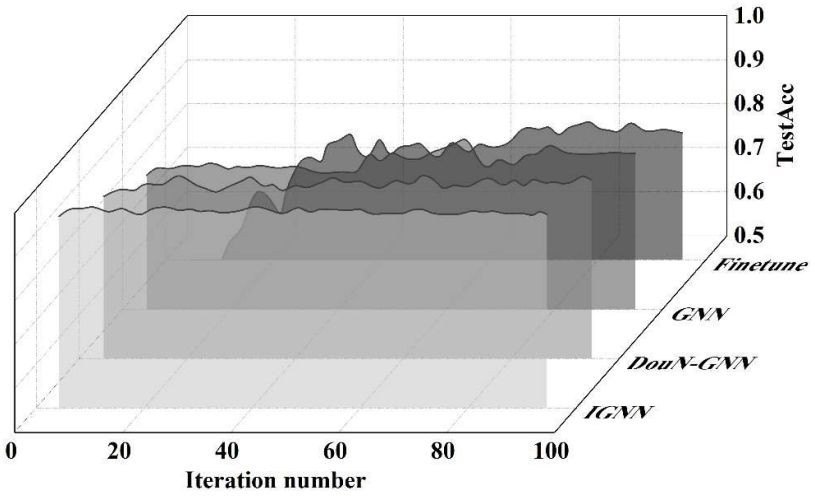


Fig. 4. Test accuracy in Omniglot data set

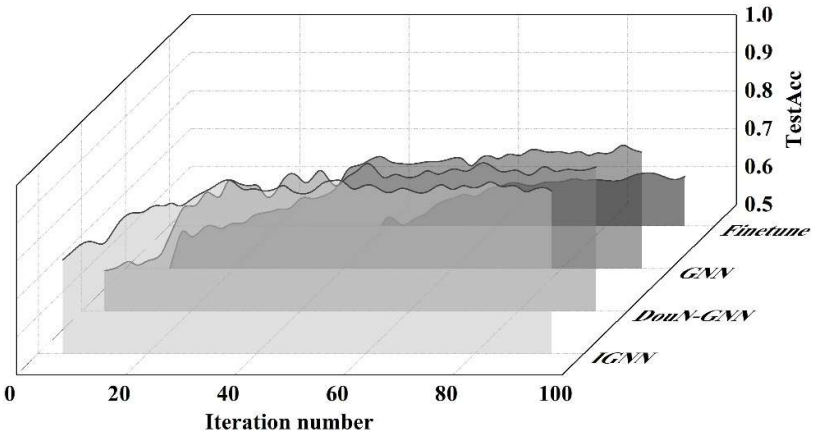


Fig. 5. Test accuracy in miniImageNet data set

Figure 6 shows the overfitting rate of the IGNN model in this paper, from which it can be seen that the overfitting rate of the model on both the Omniglot dataset and miniImageNet dataset is stable in the range of 0.95-1.00, which indicates that the training process of the experiments in this paper effectively avoids the occurrence of overfitting problems. In conclusion, this paper's optimization algorithm, the IGNN model, has faster training speed, higher accuracy, and stronger robustness in image recognition.

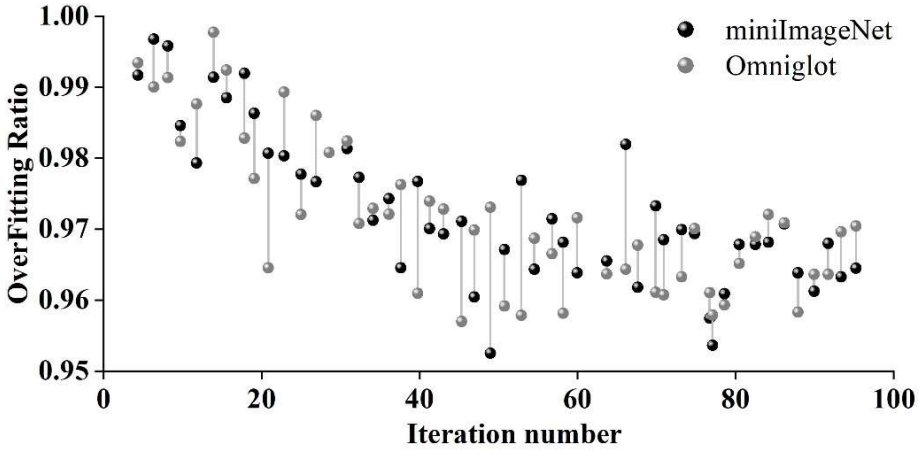


Fig. 6. Overfitting ratio of IGNN model

4. Optimize the application of image recognition algorithms

4.1. Agricultural weed identification

4.1.1. Experimental methods:

1) Data source and preprocessing. The data used in this study is the publicly available dataset DeepWeeds, which was created for the task of automatic weed identification and classification and contains 16,083 images of agricultural areas with high complexity, mainly nine different weed species native to Australia with neighboring flora, including apple, serpentine, balsam, acacia, siamese grass, bittercress, rubber vine, bermudagrass, and pussywort, a total of nine categories of agricultural weed images. This study adopted the practice of randomly removing 50% of the negative weed samples, resulting in a total final sample size of 10,364 images. Then, the samples were randomly divided into three datasets: 4146 images in the training set, 3110 in the validation set, and 3108 in the test set. The image size is set to 128 × 128, and preprocessing such as center cropping, random horizontal flipping,

and normalization are performed.

2) Experimental environment. This experiment was run in September 2023 by a college of computer science and technology using a server on Autodl; all experiments are really developed on the same server. The experiment uses Ubuntu 20.04, the CPU model is 12 vCPU Intel Xeno Platinum8255C CPU@2.50GHz, the GPU model is RTX 2080Ti, the model running framework is PyTorch 1.11.0, and the Cuda version is 11.3. The network experiment parameters are set as follows: the optimizer uses AdamW, learning rate adjustment uses the want return strategy, and in the early stage of model training, warm-up is used to warm up the learning rate to accelerate the convergence speed of the model. The number of model iterations is 150, the batch size is 8, the initial learning rate is 0.001, the weight decay method is used to suppress overfitting, and the decay coefficient is set at 0.05.

3) Evaluation indexes. This study selects the accuracy rate, recall rate, precision rate, and F1 score as the evaluation methods to judge the model's goodness. Generally speaking, the higher the accuracy, precision, recall, and F1 score values, the better the model's recognition and classification performance.

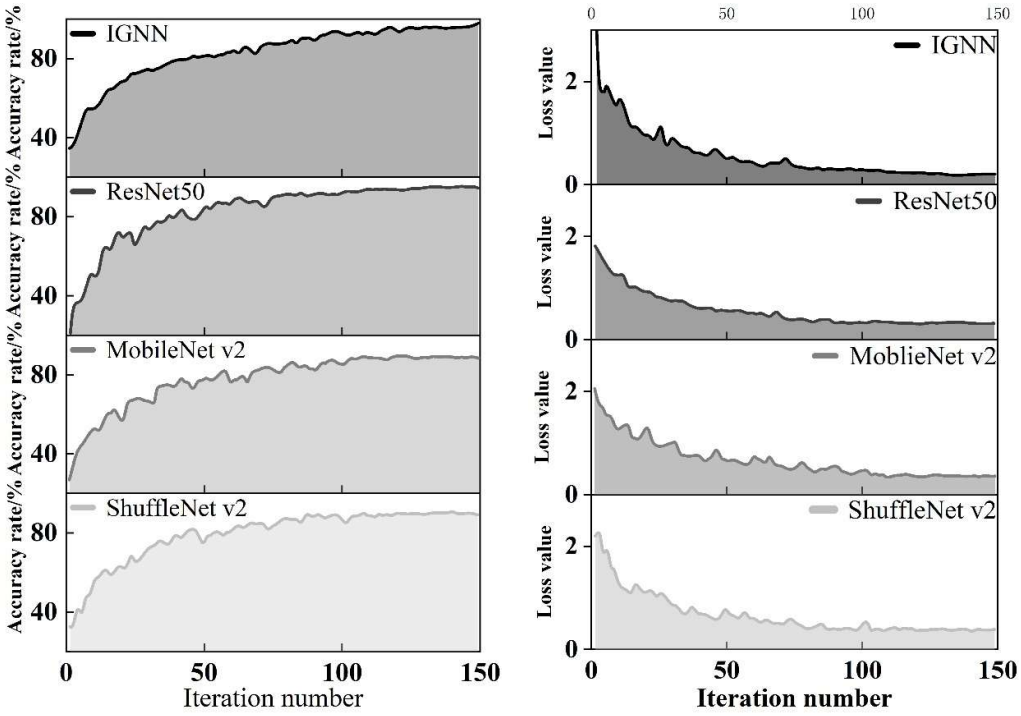
4.1.2. Analysis of experimental results:

1) Model performance analysis. In order to verify the effectiveness of the IGNN model in agricultural weed recognition, 150 iterations were performed under certain conditions. The accuracy and Loss curves were compared with three typical image recognition networks on the validation set. The results are shown in Fig. 7, where Fig. 7(a) compares the accuracy rate of the different models, and Fig. 7(b) compares the Loss values of the validation set among the different models. From Fig. 7(a), it can be seen that the model in this study has the highest accuracy of 98.39%. resNet50 reaches 94.13%, while ShuffleNet v2 and MobileNet v2 are close to 90.00%. As seen from Fig. 7(b), the Loss curve of the validation set of this study's model is smoother than the other three models, indicating a better model fit. Therefore, the model constructed in this study is best trained on the validation set compared to the mainstream classical models.

The model in this study uses accuracy, precision, recall, and F1 score to illustrate the model's performance, and the results are shown in Table 3. As shown in Table 3, the model of this study has the highest values for accuracy, precision, recall, and F1 score, and the recognition accuracy is about 4.26 percentage points higher than ResNet50. It shows that the model of this study is a good fit for the data.

Table 3. Test set accuracy results

Model	A/%	P/%	R/%	F1/%
ShuffleNet v2	88.37	87.30	87.69	87.53
MobileNet v2	89.43	87.45	88.21	88.68
ResNet50	94.13	89.08	91.28	91.08
IGNN	**98.39**	**94.95**	**96.59**	**96.47**



(a) Accuracy of different models (b) Verification set loss value

Fig. 7. Model performance results

2) Confusion matrix. The classification results of 3108 images of the test set were tested using the trained weights of the validation set, and the results of the confusion matrix for the 9 categories of weeds are shown in Fig. 8, where 1-9 represent the 9 categories of weeds such as apple, respectively. As seen from Fig. 8, the number of images predicted incorrectly for most weed categories is relatively small, with more misidentifications for 2 categories, apple and snakeweed. Apple had 8 images misidentified as negative grass and 7 images misidentified as snakeweed. Snake grass had 9 images misidentified as apple. This is because these 2 categories of image features are more similar, resulting in more mutual recognition errors for the 2 categories compared to the others. Overall, the IGNN model can achieve the desired results in terms of classification accuracy for the 9 categories of weeds.

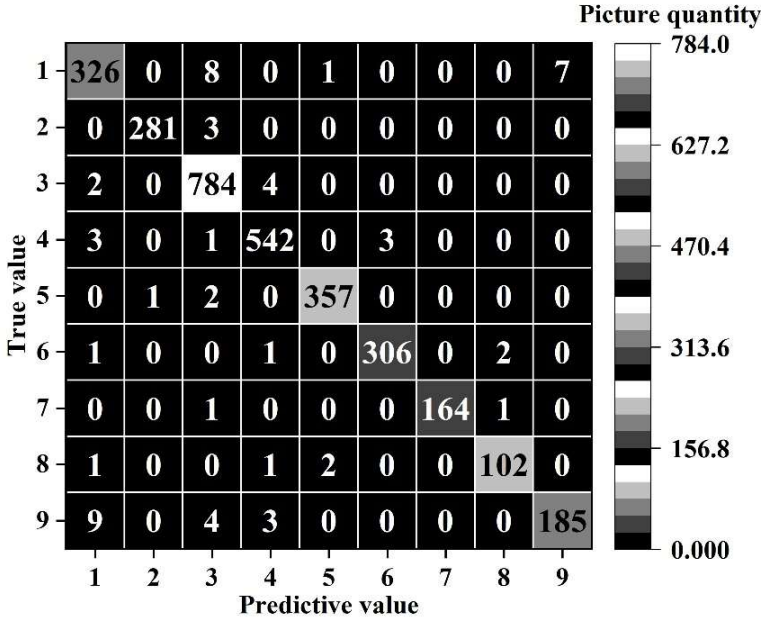


Fig. 8. The confusion matrix of weeds

4.2. Natural resource enforcement

4.2.1. Data set construction and processing. In this section, the research achieves the purpose of timely detection of illegal construction behaviors by detecting tower-based surveillance video images. Therefore, before the model training, it is first necessary to clearly define the construction behavior objects detected by the model to accurately label the target objects that meet the definition in the training samples and improve the quality of the training samples. Based on the in-depth analysis of various types of suspected illegal construction behaviors, the study selects ten types of construction behaviors as the detection objects of the target detection model in this paper.

The training sample data in this paper are obtained from the real images taken by the tower base surveillance camera in a district and preprocessed by pre-screening the original images, labeling them, and then producing a standard dataset format for generating target detection. Finally, through the screening of error images and black night images, a total of 26847 sample images that meet the training requirements of this paper's construction behavior target detection model are obtained. The statistics of the annotation results of various types of construction behaviors are shown in Table 4, and a total of 80,200 annotation results are obtained by annotating ten types of construction behaviors on the preprocessed tower-based surveillance images.

Table 4. Marked result statistics

Serial number	Label name	Number of notes
1	Dustproof	18746
2	Building	9632
3	Crane	8087
4	Excavator	8132
5	Tower-crane	12079
6	Loader	6273
7	Moraine	3582
8	Brick	594
9	Bunk-house	13038
10	Dump-tuck	37

4.2.2. Model training. In this paper, the Detectron2 framework is used for IGNN model construction and training. The base learning rate of the model is 0.003, the number of iterative training of the model is 300,000 times, and the sample dataset is augmented with random cropping and random rotation of data. Each image generates 503 Proposal results for RPN during the training and prediction process, so energy saving ensures model prediction accuracy and model training and prediction efficiency. The training Loss results of the model are shown in Fig. 9, from which it can be seen that the model loss function Loss is in a decreasing trend and finally stabilized near 0.125.

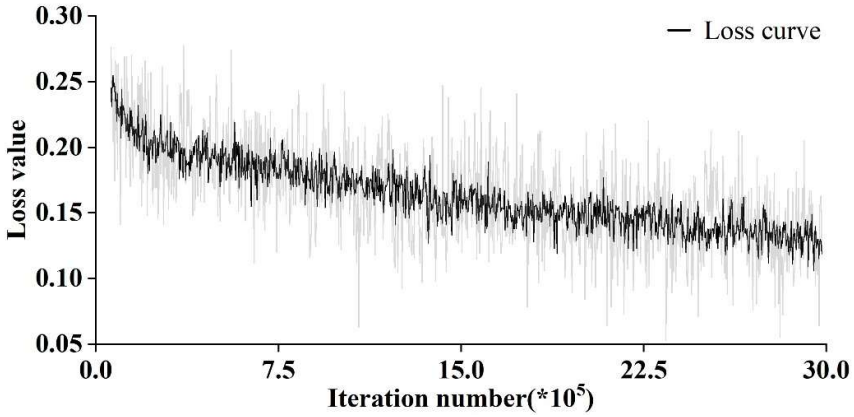


Fig. 9. Model training loss

In this paper, we will use four evaluation indexes, namely, accuracy, recall, AP, and mAP, to evaluate the precision of construction behavior image recognition obtained by IGNN model training, and the evaluation results obtained are shown in Table 5. It can be seen from the table. The results of each index of the model are above 90%, and the recognition accuracy reaches 98.74%. It shows that the IGNN model can be effectively used in the image recognition of construction behavior.

Table 5. Model accuracy evaluation results

Evaluation index	Evaluation result(%)
A	98.74
R	91.38
AP	90.29
mAP	93.46

4.2.3. Analysis of results. In this section, based on the analysis of various types of suspected illegal construction behaviors, ten types of construction behaviors with broader coverage are selected as the image recognition objects of the IGNN model in this paper, followed by pre-processing and image annotation of more than 80,000 tower monitoring images obtained to obtain training sample data that meets the training standards of the model, and then constructing the IGNN model training environment based on the Detectron2 framework development. The model training environment is developed and constructed based on the Detectron2 framework after 300,000 iterations of training to obtain the construction behavior image recognition model. Finally, the accuracy of the image recognition model of suspected illegal construction behavior obtained by training is evaluated by using the test dataset. Combined with the actual detection effect, most of the construction behaviors in the test images that meet the definition of this paper can be effectively identified, so it can be concluded that the algorithm trained based on the IGNN model in this paper can meet the requirements of the actual image recognition of suspected illegal construction behaviors.

4.3. Video Surveillance

4.3.1. Experimental methods:

1) Data Selection and Preprocessing. Data set selection is crucial for studying video surveillance image recognition technology. The experiments use video surveillance datasets from different scenarios, including city street scenes, traffic intersections, shopping malls, and factory interiors. These datasets include video frames with various weather conditions, lighting variations, and target sizes. The experiment performs the following steps in data preprocessing: first, the images are calibrated and corrected to ensure viewing angle and scale suppression. Second, data enhancement, including random rotation, cropping, blurring, and noise injection, is performed to simulate various situations in real surveillance scenes. Finally, the dataset is divided into training, validation, and test sets to ensure the effectiveness of model training and performance evaluation.

2) Model training and optimization strategy. Model training is divided into two stages. We want to use the dataset for initial training and then fine-tune the model to a specific video surveillance task. For model optimization, the experiments use the stochastic gradient descent method to optimize the model. To avoid overfitting, the experiments apply batch regularization and early stopping strategies. In addition, the convergence speed and performance of the model are improved by exploring different learning rate scheduling strategies.

4.3.2. Experimental results and analysis:

1) Comparison of recognition accuracy in different scenes. To compare the recognition accuracy in different scenarios, 3000 frames of images were randomly selected from multiple video surveillance scenes covering different weather conditions, lighting variations, and target sizes. Each image was manually labeled to indicate the target category in the image.

In the experiments, three different models are used for recognition, including ResNet50, MobileNet v2, and the IGNN model in this paper. The models are all trained in the same hardware environment, and their performance is evaluated on the same test dataset. The recognition accuracies of different models in different scenarios are shown in Table 6. From Table 6, it can be seen that the IGNN model

exhibits the highest recognition rates in city street scenes, traffic intersections, shopping malls, and factory interiors, with 96.37%, 94.52%, 95.38%, and 92.17%, respectively. The ResNet50 model, although it performs a little bit less well in these scenarios, still has high recognition accuracies, with 91.53%, 89.82%, 90.82%, 91.53%, and 92.17%, respectively, for the different scenarios, 89.82%, 90.08%, and 90.42%. The MobileNet v2 model exhibits the worst recognition accuracy, 85.48%, 82.39%, 84.37%, and 80.08% in different scenarios, respectively. This indicates that the IGNN model in this paper has an advantage in processing temporal information in video surveillance data.

Table 6. The identification accuracy of different models in different scenarios

Model	Accuracy of identification in different scenarios/%			
	Street view	Traffic junction	Mall	Factory interior
IGNN	96.37	94.52	95.83	92.17
ResNet50	91.53	89.82	90.08	90.42
MobileNet v2	85.48	82.39	84.37	80.08

2) Impact analysis of robustness and lighting conditions. Four typical lighting conditions are selected for the experiment: bright daytime, sunset, street lighting at night, and darkness at night with no light source. Under each lighting condition, different models are used for image recognition, and each case's recognition accuracy is recorded. The recognition accuracies of different models under different lighting conditions are shown in Table 7. It is clear from Table 7 that all models show good recognition accuracy under bright daytime conditions, with the IGNN, ResNet50, and MobileNet v2 models achieving 96.73%, 94.38%, and 89.25%, respectively. However, when the lighting conditions become more challenging, both ResNet50 and MobileNet v2 models show a significant decrease in accuracy, and only the IGNN model shows a smaller decrease of 9.76% in recognition accuracy under completely dark lighting conditions. This indicates that the IGNN model still performs well under poor lighting conditions.

Table 7. Identification accuracy of different models under different lighting conditions

Model	Accuracy of identification under different lighting conditions/%			
	Brightness	Sunset	Lighting	Darkness
IGNN	96.73	94.26	89.72	86.97
ResNet50	94.38	87.53	77.48	70.83
MobileNet v2	89.25	81.28	72.03	64.26

5. Conclusion

In this paper, three optimization modules are added to the two-node graph neural network DouN-GNN model to solve the problem of its low overall recognition accuracy. The optimized IGNN model is applied to real-world scenarios such as agricultural weed recognition, natural resource enforcement, and video surveillance.

1) The model's recognition accuracy for agricultural weeds is as high as 98.39%, which is 4.26% higher than that of the next best-performing ResNet50 model. And the model has the highest accuracy, precision, recall and F1 score values. Meanwhile, the overall classification and recognition accuracy of the IGNN model in classifying and recognizing nine types of agricultural weeds, such as apple and snakeweed, also reached the ideal state. It indicates that the IGNN model performs well in agricultural weed recognition.

2) In the model of construction behavior image recognition, the results of all indicators are more than 90%, and the recognition accuracy is as high as 98.74%. Combined with the actual detection results found in the test image, they align with the definition of this paper's construction behavior most can

be effectively recognized. The image recognition based on the IGNN model in this paper can promptly detect all kinds of illegal construction behaviors in construction projects, fully reflecting the trend of intelligence in natural resources law enforcement.

3) The IGNN model shows the highest recognition rate in video surveillance image potential recognition in different scenarios, with a recognition accuracy of 96.37%, 94.52%, 95.38%, and 92.17% in urban street scenes, traffic intersections, shopping malls, and internal scenes of factories, respectively. Meanwhile, the performance of IGNN remains stable under different lighting conditions, and the recognition accuracy of the IGNN model in completely dark lighting conditions decreases by only 9.76% compared with that in bright lighting conditions. It shows that the image recognition algorithm based on the IGNN model can greatly meet the requirements of video surveillance in different scenes and lighting conditions.

References

[1] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 32(1), 4-24.

[2] Sarlin, P. E., DeTone, D., Malisiewicz, T., & Rabinovich, A. (2020). Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4938-4947).

[3] Knyazev, B., Taylor, G. W., & Amer, M. (2019). Understanding attention and generalization in graph neural networks. Advances in neural information processing systems, 32.

[4] Shen, Y., Li, H., Yi, S., Chen, D., & Wang, X. (2018). Person re-identification with deep similarity-guided graph neural network. In Proceedings of the European conference on computer vision (ECCV) (pp. 486-504).

[5] Yuan, H., Yu, H., Wang, J., Li, K., & Ji, S. (2021, July). On explainability of graph neural networks via subgraph explorations. In International conference on machine learning (pp. 12241-12252). PMLR.

[6] Keriven, N., & Peyré, G. (2019). Universal invariant and equivariant graph neural networks. Advances in Neural Information Processing Systems, 32.

[7] Holzinger, A., Malle, B., Saranti, A., & Pfeifer, B. (2021). Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI. Information Fusion, 71, 28-37.

[8] Wang, W., Lu, X., Shen, J., Crandall, D. J., & Shao, L. (2019). Zero-shot video object segmentation via attentive graph neural networks. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9236-9245).

[9] Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., & Hoffmann, H. (2019). Explainability methods for graph convolutional neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10772-10781).

[10] Du, S. S., Hou, K., Salakhutdinov, R. R., Poczos, B., Wang, R., & Xu, K. (2019). Graph neural tangent kernel: Fusing graph neural networks with graph kernels. Advances in neural information processing systems, 32.

[11] Chen, Y., Wu, L., & Zaki, M. (2020). Iterative deep graph learning for graph neural networks: Better and robust node embeddings. Advances in neural information processing systems, 33, 19314-19326.

[12] Wieder, O., Kohlbacher, S., Kuenemann, M., Garon, A., Ducrot, P., Seidel, T., & Langer, T. (2020). A compact review of molecular property prediction with graph neural networks. Drug Discovery Today: Technologies, 37, 1-12.

[13] Zhang, C., Song, D., Huang, C., Swami, A., & Chawla, N. V. (2019, July). Heterogeneous graph neural network.

In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 793-803).

[14] Wu, L., Chen, Y., Shen, K., Guo, X., Gao, H., Li, S., ... & Long, B. (2023). Graph neural networks for natural language processing: A survey. Foundations and Trends® in Machine Learning, 16(2), 119-328.

[15] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., & Grohe, M. (2019, July). Weisfeiler and leman go neural: Higher-order graph neural networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 4602-4609).

[16] Chen, Z. M., Wei, X. S., Wang, P., & Guo, Y. (2019). Multi-label image recognition with graph convolutional networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5177-5186).

[17] Li, R., Tapaswi, M., Liao, R., Jia, J., Urtasun, R., & Fidler, S. (2017). Situation recognition with graph neural networks. In Proceedings of the IEEE international conference on computer vision (pp. 4173-4182).

[18] Gong, L., & Cheng, Q. (2019). Exploiting edge features for graph neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9211-9219).

[19] Kim, J., Kim, T., Kim, S., & Yoo, C. D. (2019). Edge-labeling graph neural network for few-shot learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11-20).

[20] Hong, D., Gao, L., Yao, J., Zhang, B., Plaza, A., & Chanussot, J. (2020). Graph convolutional networks for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 59(7), 5966-5978.

[21] Zhao, T., Zhang, X., & Wang, S. (2021, March). Graphsmote: Imbalanced node classification on graphs with graph neural networks. In Proceedings of the 14th ACM international conference on web search and data mining (pp. 833-841).

[22] Tan Li,Lv Xinyue,Wang Ge & Lian Xiaofeng. (2023). UAV image object recognition method based on small sample learning. Multimedia Tools and Applications(17),26631-26642.

[23] Shuai Zhang,Yuanze Du,Yingwang Zhao & Lifu Zhou. (2024). Image Recognition of Mine Water Inrush Based on Bilinear Convolutional Neural Network with Few-Shot Learning. ACS omega(10),12027-12036.

[24] China University of Petroleum (East China),China University of Petroleum (East China),Yunnan University & Faculty of Science and Technology, University of Macau. (2019). Ensemble p-Laplacian Regularization for Scene Image Recognition. Cognitive Computation(6),841-854.

[25] Farouk R.M,Badr E.M & SayedElahl M.A. (2014). Recognition of CDNA Microarray Image using Feedforward Artificial Neural Network. International Journal of Artificial Intelligence & Applications(5),21-31.