Computers and Artificial Intelligence；Vol.3, No.2; 2026
E-ISSN: 3006 -3728 P-ISSN:3006-371X
DOI: https://doi.org/10.70267/cai.26v3n2.4158
Published by: Zeus Press

Research on a Weakly Supervised Algorithm Based on Few-Shot Learning

Yufan Wu*

College of Smart Agriculture (College of Artificial Intelligence), Nanjing Agricultural University (Pukou Campus), Nanjing 210032, China

**Corresponding author: Yufan Wu.*

Abstract

To address the issues of weakly supervised learning (WSL) such as limited feature diversity, poor adaptability to new scenarios, and the incomplete intra-class feature coverage and low accuracy in complex scenarios inherent to few-shot learning (FSL), this paper proposes a weakly supervised algorithm based on few-shot learning. The algorithm first constructs a rule-based system through five core modules to generate approximately accurate pseudo-labels for large-scale unlabeled data. Subsequently, an intra-class feature expansion optimization strategy is designed to mine effective features using methods such as weighted distance metrics, optimizing the feature space and reducing the risk of model overfitting. By fully exploiting the effective information in data and optimizing the structure of the feature space, the algorithm compensates for the lack of intra-class feature diversity in FSL, effectively alleviates overfitting, and improves classification accuracy and generalization in complex scenarios. Experimental results demonstrate that, compared with existing literature, the proposed few-shot feature expansion algorithm under weak supervision achieves at least a 1.18% improvement in accuracy, 2.30% in precision, 1.19% in recall, and 1.76% in F1 score. For strongly supervised algorithms, the improvements are at least 4.55%, 4.49%, 4.60%, and 4.55%, respectively. The proposed approach provides a novel solution for model training under data scarcity conditions.

Keywords

few-shot learning, weakly supervised learning, feature expansion, joint training

1. Introduction

With the development of intelligent and automated systems, few-shot learning (FSL) alone is insufficient to achieve generalized training in scenarios requiring massive labeled samples, especially in large-scale and diverse real-world data environments. The scarcity of training samples leads to unstable feature representation and limited generalization performance [1, 2]. Traditional supervised learning relies heavily on large numbers of labeled samples, which incurs high labeling costs, is time-consuming, and labor-intensive. Moreover, obtaining labeled data in specific domains is extremely difficult, significantly increasing acquisition costs and limiting the application of supervised learning models in fields such as healthcare and finance.

To address the feature sparsity and limited generalization of weakly supervised learning (WSL), FSL can be introduced. FSL substantially reduces the requirement for labeled data, enabling effective model training and inference in data-scarce scenarios. Its rapid adaptation allows models to learn core features of new domains from minimal labeled data, thereby compensating for the limited feature generalization of WSL [3]. This capability enables models to extract effective feature representations and decision boundaries from very few samples, facilitating swift adaptation to new task categories, which is critical for dynamic, open-domain data scenarios [4]. In addition, FSL reduces the dependency of traditional deep learning on large-scale datasets, lowering the data threshold for training and extending the applicability of WSL to data-scarce domains [5]. However, FSL faces challenges when the number of classes is high and the number of samples per class is limited. Few-shot classes provide insufficient intra-class samples to cover feature diversity, making it difficult for models to accurately capture common intra-class characteristics, which negatively affects classification accuracy [6]. Furthermore, when the number of classes significantly exceeds the samples per class, models often overfit, performing well on the training set but generalizing poorly to unseen test data [7]. FSL methods based on convolutional neural networks (CNNs) and graph neural networks (GNNs) show decreasing performance as the number of classes increases and intra-class samples decrease. This is because the complexity of decision boundaries grows exponentially with the number of classes, while the sparse intra-class samples provide insufficient support for accurate boundary estimation [8]. Balancing overfitting and effective feature representation remains a core challenge in few-shot image classification. Existing FSL strategies under deep neural networks (DNNs) still leave significant room for accuracy improvement on complex datasets. Current approaches often fail to fully exploit the effective information in high-dimensional complex features, limiting model performance [9]. Weakly supervised learning can be leveraged to mitigate these limitations. Nevertheless, WSL alone suffers from sparse features and limited generalization, while FSL alone struggles with insufficient intra-class coverage, overfitting, and reduced accuracy in complex scenarios. Both approaches have clear limitations when applied independently, and existing attempts to combine them have not fully resolved these core issues. Additionally, current algorithms face challenges such as poor adaptation across multiple weak supervision types, high computational complexity, and limited theoretical analysis of generalization. Existing few-shot graph anomaly detection methods rely on meta-training tasks with abundant labeled samples, which contradicts real-world scenarios where labeled data in meta-training tasks is often below 0.1%. This discrepancy between meta-training and meta-testing tasks leads to significant overfitting [10].

Motivated by these challenges, this paper proposes a weakly supervised algorithm based on FSL. The algorithm constructs a rule-based system to generate approximately accurate pseudo-labels, providing weak supervision signals for large-scale samples, and integrates the weakly supervised model into an overall training framework. This approach effectively expands the usable training sample size and allows learning more robust data distribution features under weak supervision constraints, reducing dependence on high-quality labeled data and alleviating overfitting, thereby improving generalization in practical applications. Considering the insufficient intra-class feature coverage, overfitting, and reduced recognition accuracy of FSL models in complex scenarios, we further design a few-shot feature expansion optimization algorithm. This method extends the intra-class feature space through weak supervision to enhance model representation and robustness. Specifically, it replaces precise labels with rules, builds a rule library for large-scale unlabeled data, and identifies target samples and their feature vectors based on a semantic–syntactic matching mechanism, using them as supplementary features in the FSL process.

The main contributions of this paper are as follows:

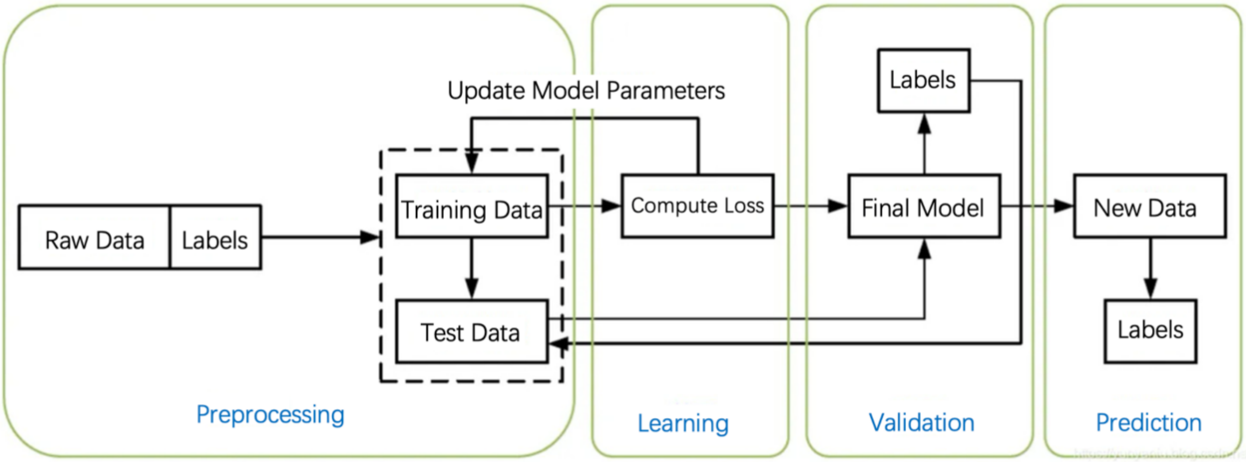
1) A rule-based weakly supervised learning framework is proposed to reduce reliance on exact labels through pseudo-label generation.
2) An intra-class feature expansion algorithm is designed to enhance feature representation using unlabeled data.
3) A joint training approach integrates strong and weak supervision signals, effectively reducing overfitting and improving classification accuracy in complex scenarios.

2. Related Techniques

2.1 Weakly Supervised Learning

Weakly supervised learning (WSL) refers to training models using incomplete, imprecise, or inaccurate weak labels to address the core issue of scarcity and high cost of high-quality annotations. This technique effectively combines supervised and unsupervised learning methods by using both labeled and unlabeled data during training [12]. Currently, mainstream weakly supervised object detection algorithms are implemented based on convolutional neural networks (CNNs) [13-15]. WSL first preprocesses the collected weakly labeled data, converting weak label information into supervisory signals recognizable by the model. Using deep neural networks such as fully convolutional networks (FCNs) and CNNs as the base model, the preprocessed weakly labeled data are input into the model for initial training [16, 17]. During training, techniques such as multiple-instance learning, constraint optimization, and saliency detection are combined to establish associations between image annotations and pixel-level semantics, compensating for the missing information in weak labels. The workflow is illustrated in Figure 1.

Figure 1: Workflow of Weakly Supervised Learning

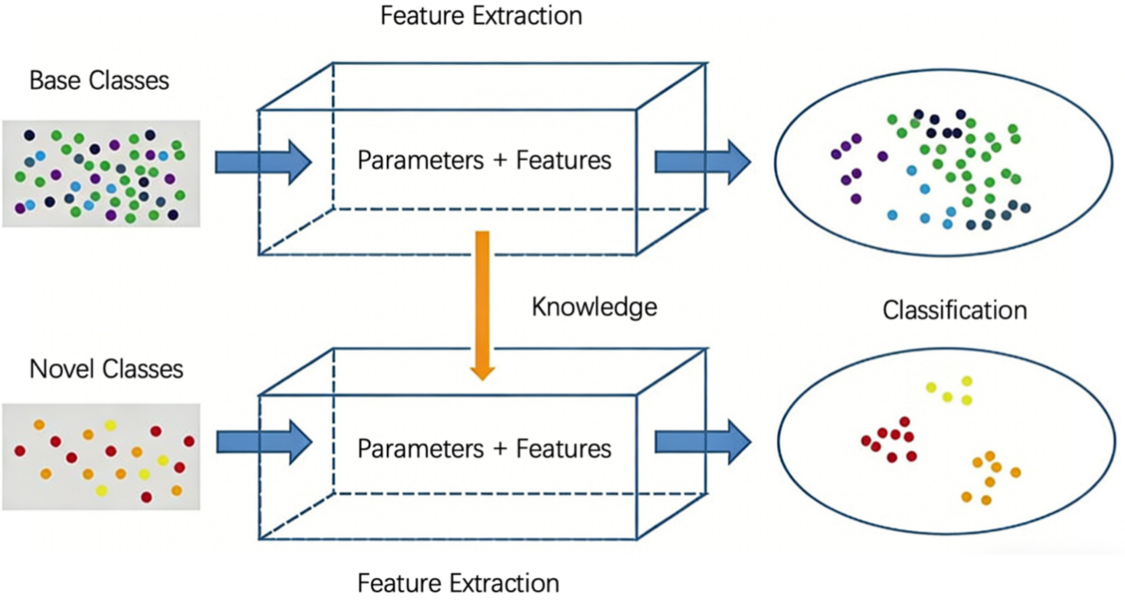


As shown in Figure 1, the process begins with data preparation, which requires collecting two types of core data: (1) a small amount of weakly labeled data with limited annotation accuracy, and (2) a large volume of completely unlabeled data. Both data types should cover the intended application scenarios of the model, and their quality should meet basic preprocessing requirements. Next, weakly labeled data undergo preprocessing, which involves cleaning the data to remove blurred, incorrect, or redundant samples, and standardizing the remaining valid data to ensure consistency and processability. After preprocessing, it is checked whether the weakly labeled data have been converted into effective supervisory signals recognizable by the deep neural network. If not, the preprocessing step is repeated and optimized until the data are successfully transformed. The processed data carrying valid supervisory signals are then fed into deep neural networks such as CNNs or FCNs to initiate initial model training. During training, it is necessary to determine whether the model has established associations between image annotations and pixel-level semantics. If these associations have not yet been formed, saliency detection techniques are introduced to enhance the extraction of critical image regions, multiple-instance learning is applied to uncover latent patterns from multiple data samples, and constraint optimization is used to control training errors while continuously adjusting model parameters. After multiple rounds of training and parameter optimization, once the model meets the predefined performance criteria, training is completed. Subsequently, when new data are introduced, the trained model can directly process them and output corresponding predictions.

2.2 Few-Shot Learning

Few-shot learning (FSL) refers to learning methods to solve tasks from a limited number of samples [3]. The data are divided into a training set, validation set, and test set, ensuring that the classes in the test tasks are previously unseen by the model to evaluate generalization capability. FSL tasks are constructed as N-way K-shot tasks. The model is trained via meta-learning, learning across multiple meta-tasks. In each training iteration, a meta-task is randomly sampled to generate a support set and a query set. The model learns class features from the support set and computes classification loss on the query set. Model parameters are updated across multiple meta-tasks to achieve high accuracy.

Figure 2: Workflow of Few-Shot Learning



As shown in Figure 2, the core idea of prototypical networks—computing the mean of support set features to construct class prototypes—is widely applied in FSL tasks. Essentially, this approach uses metric learning to aggregate and discriminate class features, representing a classic paradigm for few-shot classification [18]. The core workflow of FSL begins with data partitioning, dividing all available data into training, validation, and test sets. A key point is that the test set must contain classes completely unseen during training to accurately evaluate the model's ability to recognize novel classes. Next, specialized few-shot tasks are constructed, typically involving a small number of classes and few samples per class (i.e., N-way K-shot tasks). For each training iteration, a task is randomly sampled from the task pool to expose the model to different class combinations, rather than repeatedly using the same task. Within each task, the data are split into two subsets: a small set of learning samples called the support set, and samples for classification called the query set. The support set provides the basis for the model to learn, while the query set evaluates the learning performance. During training, the model first uses the support set to learn the core features of each new class, effectively acquiring the key characteristics for class recognition. Then, classification is performed on the query set to compute classification loss, which measures errors and is used to adjust model parameters to gradually improve classification performance. This process is repeated across multiple few-shot tasks, allowing the model to learn from different class combinations and sample scenarios, continuously optimizing performance. Finally, the model is evaluated on the reserved validation and test sets to measure its recognition ability on novel classes, completing the FSL process.

In the meta-training process of FSL, feature learning focuses on extracting discriminative class features from the support set [19, 20]. The formula is as follows:

$$\boldsymbol{c_k} = \frac{1}{n_k}\sum_{i=1}^{n_k} f_\phi(\boldsymbol{x_{k,i}}) \tag{1}$$

In this context, c_k represents the feature prototype of the k -th class, obtained by averaging the features of all support set samples in that class; n_k denotes the number of samples in the support set for the k -th class; and $f_\phi(\cdot)$ is the feature extraction network parameterized by ϕ, which maps the input sample $x_{k,i}$ to a high-dimensional feature space.

The classification loss aims to minimize the distance between query set samples and the prototypes of their true classes, while maximizing the distance to prototypes of other classes, thereby improving classification accuracy. The formula is as follows:

$$\mathcal{L} = -\frac{1}{N_q}\sum_{j=1}^{N_q} \log\left(\frac{\exp\left(-d\left(f_{\phi}(\boldsymbol{x}_j),\boldsymbol{c}_{y_j}\right)\right)}{\sum_{k=1}^{N} \exp\left(-d\left(f_{\phi}(\boldsymbol{x}_j),\boldsymbol{c}_k\right)\right)}\right) \tag{2}$$

In this context, N_q denotes the total number of query set samples; $d(\cdot,\ \cdot)$ the Euclidean distance function; y_j the true class label of query sample x_j; and N the total number of classes in the few-shot task, i.e., the number of classes in an N-way task.

The core challenge in weakly supervised learning lies in the insufficient quality of supervisory signals. Few-shot learning, by leveraging the efficient learning capability of a small number of accurate samples, can mitigate the misleading effects of noisy weak labels. However, using such methods alone may still be limited by sample quality. Minor errors during label design can cause overfitting in few-shot learning, which in turn may misguide weakly supervised learning, amplifying the impact of noisy labels.

2.3 Research Motivation

Considering the high cost and time-consuming nature of obtaining large numbers of accurately labeled samples, especially in few-shot learning, traditional FSL approaches introduce a small number of labeled samples alongside a large amount of unlabeled data to address annotation sparsity. However, these methods often face label noise and inconsistency. The generation of weak supervision labels typically relies on heuristic rules and weak-label libraries, which are neither precise nor consistent, leading the model to learn incorrect patterns and exacerbating overfitting. To address these issues, this paper proposes a label-independent training strategy. By constructing a rule-based system to generate approximately accurate labels, weak supervision signals are provided for large-scale model samples. Incorporating the weakly supervised model into the training process not only expands the available training data but also allows learning more robust distribution patterns, reducing reliance on precisely labeled data and alleviating overfitting, thereby enhancing the practical generalization ability of the model. Few-shot learning is designed to model and classify new classes with only a few labeled samples. However, in complex scenarios, it faces challenges such as insufficient feature coverage, overfitting on training data, and reduced recognition accuracy. To address these issues, this paper proposes an intra-class feature expansion optimization algorithm. Using weak supervision, the algorithm expands the intra-class feature space to enhance model representation and robustness. Specifically, it replaces precise labels with rules, constructs a rule library over large-scale unlabeled data, and identifies target samples and their feature vectors through semantic–syntactic matching. These features are incorporated into the original few-shot learning process. This approach not only alleviates the scarcity of features in few-shot learning but also improves the model's perception of intra-class semantic–syntactic information, thereby enhancing classification performance and generalization across different environments.

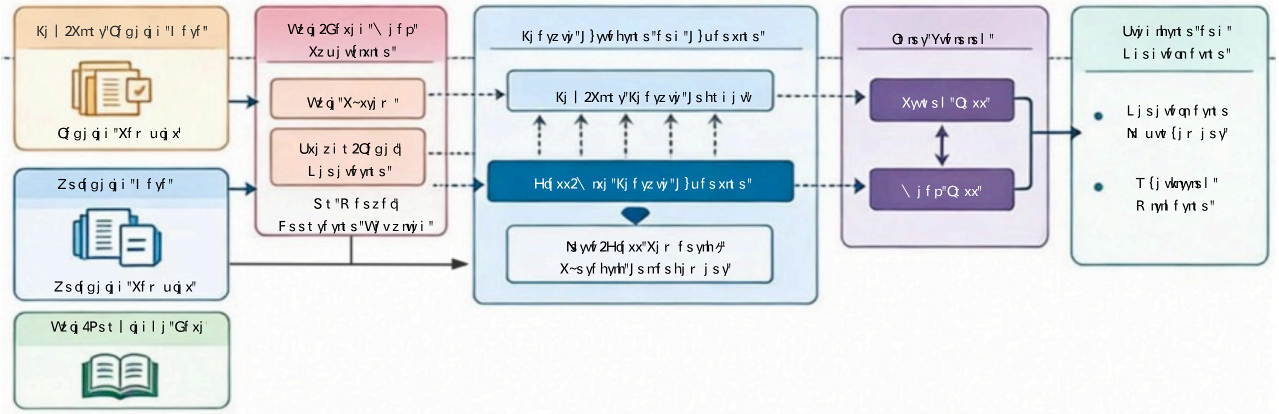
3. A Weakly Supervised Algorithm Based on Few-Shot Learning

3.1 System Framework

This study integrates few-shot learning techniques and proposes a rule-based weak supervision signal generation framework to effectively leverage unlabeled data and reduce dependence on manual labeling. Within this framework, weak supervision signals are generated through a rule system to assist the few-shot learning model in effective training under limited labeled data. The framework workflow is systematically illustrated in Figure 3, comprising five core modules: data input, rule system and pseudo-label generation, feature extraction and expansion, joint training and optimization, and prediction and generalization.

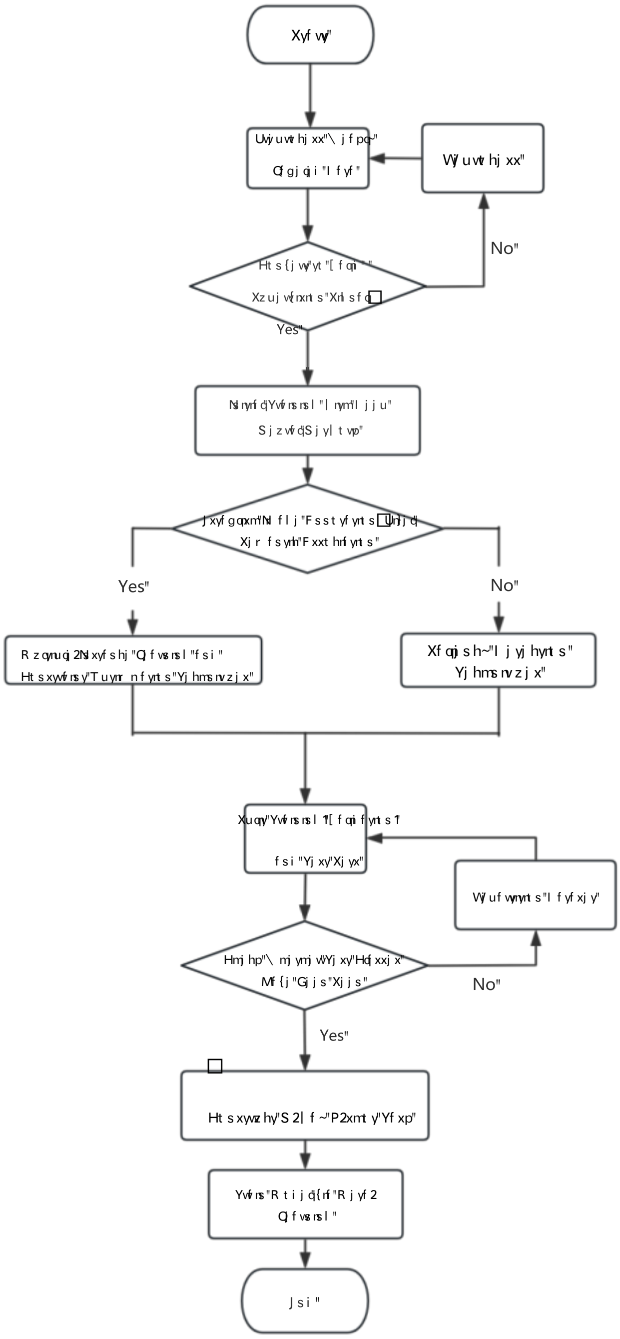
As shown in Figure 3, in the data input module, limited labeled data and a large volume of unlabeled data are first fed into the system. Unlabeled data usually constitute the majority of the dataset, while the limited labeled samples provide initial guidance for model training. The system then automatically generates pseudo-labels through a preconstructed rule library, greatly reducing the need for manual annotation and providing additional data for further training. In the rule system and pseudo-label generation module, the rule-based system performs semantic and syntactic analysis on unlabeled data to automatically generate pseudo-labels. These pseudo-labels serve as weak supervision signals, increasing the effective training data and ensuring a certain level of label consistency and rationality, reducing reliance on manual labeling. In the feature extraction and expansion module, the system employs a few-shot feature encoder to extract features from the limited labeled samples and extends these features. Specifically, intra-class feature expansion allows the model to extract latent intra-class information from unlabeled data, enriching the feature space. Additionally, semantic–syntactic enhancement enables the model to capture finer-grained semantic and syntactic information, improving feature representation capability. In the joint training and optimization module, the model combines strong-label algorithms with the weak supervision-based feature expansion module for joint training. Under this strategy, the model is further optimized under guidance from precise labels while leveraging weak labels to expand the learning scope. This joint training approach effectively alleviates overfitting and enhances the model's adaptability and stability to few-shot data. Finally, in the prediction and generalization module, the framework significantly improves classification accuracy and enhances generalization across diverse environments and scenarios. By introducing weak supervision signals, the model can better adapt to new classes and complex contexts, producing accurate predictions while avoiding performance degradation caused by overfitting. The detailed workflow is illustrated in Figure 4.

Figure 3: System Framework



As shown in Figure 4, this process leverages weakly supervised learning to compensate for the shortage of precise labels, fundamentally mitigating the overfitting problem during model training. In real-world scenarios, acquiring accurate labels is costly and time-consuming, requiring expert annotators and multiple validation steps. Consequently, high-quality labeled data are often scarce, while a large volume of unlabeled data remains unused. This workflow addresses this issue by using massive unlabeled data as the initial input and applying predefined rules to automatically generate approximately correct weak labels. Although these weak labels are not perfectly accurate, they roughly reflect the core characteristics of the data, transforming previously unlabeled raw data into usable supervisory signals for subsequent training. The weakly labeled data are not directly input into the model but are incorporated under a controlled training strategy. Since weak labels may contain noise or inaccuracies, unfiltered usage could negatively affect model performance. Therefore, the process employs scientific strategies to manage training, including multiple-instance learning to reduce the impact of individual inaccurate weak labels, constraint optimization to guide model learning via appropriate parameter ranges and loss functions, saliency detection to identify critical information for training, and establishing semantic relationships between data samples to enhance the model's understanding of underlying logic. The specific algorithm is detailed in Algorithm 1.

Figure 4: System Flowchart



Through Algorithm 1, the number of training samples is substantially expanded, and sample diversity is significantly improved. The model no longer relies solely on a few scarce precise samples but can capture stable statistical patterns across broader and more diverse sample distributions. This training paradigm, based on a large amount of weakly labeled data, enables the model to learn more generalizable and robust features, reducing over-reliance on individual precise samples and effectively mitigating overfitting, thereby enhancing model generalization. The core of strong-label design lies in guiding the model accurately via carefully designed loss functions and precision mechanisms with periodic updates. Specifically, the designed loss function reinforces intra-class compactness and inter-class separation for strong-label samples, while reducing the influence of low-confidence weak labels. The precision mechanism incorporates periodic feedback based on validation set performance to correct overfitting in labels.

Algorithm 1: Few-Shot Feature Expansion Algorithm Based on Weakly Supervised Learning

```
Input: Few-shot dataset with precise labels $D_s$ and large-scale unlabeled dataset $D_u$
Output: Expanded feature set
Feature Preprocessing: Align the feature dimensions of the datasets
Standardize the unified feature dimensions using Z-score normalization: $x^* = (x - \mu)/\sigma$
Apply saliency detection to select high-value features from samples
Assign samples to $Ds\_proc$ or $Du\_proc$ according to their origin
Weak Label Generation: Generate weak labels $y_u$ based on nearest neighbors in $D_{s\_\text{proc}}$, and compute the confidence conf($x_u$)
Feature Alignment and Enhancement: Compute the Euclidean distance between feature distributions: dist = $\|P(x_w) - P(D_{S\_proc}^{yw})\|_2$
if dist <γ (distribution shift threshold) then
    Add ($x_w$, $y_w$, conf($x_w$)) to $D_a$
end if
Feature Fusion: Merge the few-shot dataset and high-confidence weakly labeled feature set
```

3.2 Optimization Design of the Few-Shot Feature Expansion Algorithm Based on Weakly Supervised Learning

Traditional weakly supervised feature expansion methods and few-shot learning usually adopt independent, unidirectional patterns. The feature mining process in weak supervision is not constrained by target class features in few-shot learning, which may lead to mined features deviating from the core feature space of the target class. Consequently, although the dataset is expanded by increasing the number of features, the resulting feature set may have poor class discriminability and limited improvement in feature representation, potentially introducing feature redundancy and reducing classification performance. Conventional few-shot learning often uses Euclidean distance as the sole selection metric:

$$d(f_{\text{cani}}, \mu_c) = \sqrt{\sum_{k=1}^{D} \left(f_{\text{cani},k} - \mu_{c,k}\right)^2} \tag{3}$$

where f_{cani} denotes the feature vector of sample i, μ_c is the prototype mean vector of class c, $f_{\text{cani},k}$ is the k-th feature of sample i, $\mu_{c,k}$ is the k-th dimension of class c, and D is the feature dimension. However, this Euclidean-distance-based approach relies solely on the prototype mean vector and ignores intra-class feature diversity and fine-grained distribution information. Therefore, despite the increased number of features, coverage of the feature space remains insufficient, especially in complex scenarios, resulting in overfitting and reduced classification accuracy for new classes.

To address insufficient intra-class feature coverage, overfitting, and low accuracy in complex scenarios, this paper proposes an optimized few-shot feature expansion algorithm. By combining weak supervision and few-shot feature expansion, the algorithm improves the traditional feature expansion process. Specifically, intra-class feature expansion and semantic–syntactic enhancement are introduced to improve feature discriminability and representativeness, enhancing the representation capability of few-shot classes.

During feature expansion, a weighted distance metric is introduced to overcome the limitations of solely using Euclidean distance. The contribution of each feature dimension is weighted to improve feature selection:

$$d_{\text{opt}}(f_{\text{cani}}, \mu_c) = \sqrt{\sum_{k=1}^{D} w_k \left(f_{\text{cani},k} - \mu_{c,k}\right)^2} \tag{4}$$

where w_k denotes the weight of the k-th feature dimension, representing its importance in the overall feature space. The weight w_k can be dynamically adjusted based on class feature semantics or weak supervision signals.

Introducing weights allows the model to focus on important features while ignoring redundant or irrelevant ones, avoiding feature redundancy and information loss.

The intra-class feature expansion technique further optimizes the feature space. Traditional methods often rely only on intra-class means and a limited number of expanded samples, which restricts the feature space. To overcome this limitation, this paper expands features using weak supervision signals derived from unlabeled data, combined with pseudo-labels generated from semantic and syntactic rules, enhancing feature space coverage.

To further improve the accuracy of expanded features, an intra-class information mining and expansion model is designed, integrating clustering analysis and information gain strategies:

$$\mathrm{I}_{\mathrm{intra}}(\mathrm{C}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{\mathrm{j}=1}^{\mathrm{n}} \left(\mathrm{P}(\mathrm{f}_{\mathrm{cani}}) \log \quad \frac{\mathrm{P}(\mathrm{f}_{\mathrm{cani}})}{\mathrm{P}(\mathrm{f}_{\mathrm{cani,i}})} \right) \tag{5}$$

where $I_{\mathrm{intra}}(C)$ denotes the intra-class information gain, $P(f_{\mathrm{cani}})$ represents the probability distribution of the features, and $P(f_{\mathrm{cani},i})$ is the conditional probability of intra-class samples.

To address insufficient intra-class feature coverage, a class-feature-based weighted feature expansion method is introduced. This method dynamically adjusts the weight of each feature dimension, thereby giving greater emphasis to key information relevant to the class during feature expansion. The specific weight adjustment formula is as follows:

$$w_k = \frac{\sum_{i=1}^{n} |f_{\mathrm{cani},k} - \mu_{c,k}|}{\sum_{i=1}^{n} |f_{\mathrm{cani},k}|} \tag{6}$$

where w_k is the weight of the k-th feature, f $f_{\mathrm{cani},k}$ is the value of the k-th feature of sample i, and $\mu_{c,k}$ is the mean of class c in the k-th dimension. The weight for each feature dimension is determined by calculating its variability, with higher-weighted features receiving more attention during expansion, thereby reducing the influence of redundant features.

In addition to intra-class feature expansion, optimizing the differences between classes is also crucial for few-shot learning. Traditional methods rely solely on Euclidean distance to measure inter-class differences; however, in complex scenarios, this distance metric may be insufficient to capture subtle distinctions between classes. Therefore, this paper introduces a weighted inter-class distance metric, which is optimized using the following formula:

$$d_{\mathrm{inter}}(C_1, C_2) = \sqrt{\sum_{k=1}^{D} \left(w_k (f_{C_1,k} - f_{C_2,k}) \right)^2} \tag{7}$$

where $f_{C\ ,k}$ and $f_{C\ ,k}$ denote the mean feature values of classes C_1 and C_2 in the k-th dimension, respectively, and w_k is the weight coefficient of the k-th feature dimension. By weighting the differences across feature dimensions, the inter-class distance metric is optimized, enabling the model to focus more on dimensions with stronger discriminative power during feature expansion.

In weakly supervised learning, the quality of the pseudo-label generation process directly affects model performance. To improve the accuracy of pseudo-label generation, this paper incorporates an information gain method to optimize each pseudo-label generation process. Information gain is used to measure the discriminability and effectiveness of features, and the formula is as follows:

$$IG(X) = H(X) - \sum_{i=1}^{m} P(x_i) H(X|x_i) \tag{8}$$

where $IG(X)$ denotes the information gain of feature X, $H(X)$ represents the entropy of feature X, $P(x_i)$ is the probability of occurrence of sample x_i, and $H(X|x_i)$ is the conditional entropy of feature X given x_i. By maximizing the information gain, the most discriminative features can be selected during the feature expansion process, thereby improving the accuracy of pseudo-labels.

To further enhance the learning effectiveness during feature expansion, this paper designs a comprehensive loss function that simultaneously considers inter-class distance, intra-class feature coverage, and information gain. The formulation is as follows:

$$L_{\mathrm{opt}} = \lambda_1 \sum_{i=1}^{N} d(f_{\mathrm{cani}}, \mu_c) + \lambda_2 \sum_{k=1}^{D} w_k \left(f_{\mathrm{cani},k} - \mu_{c,k} \right)^2 + \lambda_3 \sum_{i=1}^{M} IG(x_i) \tag{9}$$

where L_{opt} denotes the optimized loss function, $d(f_{\text{cani}}, \mu_c)$ represents the Euclidean distance between a sample and its class center, w_k is the weight coefficient of the feature, $IG(x_i)$ denotes the information gain, and λ_1, λ_2, and λ_3 are weighting parameters used to control the relative importance of different terms. By jointly optimizing these components, the loss function guides the model during feature expansion to maximize inter-class separability, minimize intra-class variance, and enhance the effectiveness of information representation.

In summary, by introducing weighted feature expansion, inter-class distance optimization, and information gain, the feature representation capability in few-shot learning is effectively improved. Specifically, the optimized model enhances the feature space through weak supervision signals while mitigating overfitting, thereby improving classification performance in complex scenarios. The final optimization objective is formulated as follows:

$$f_{\text{final}} = \arg\min_{\theta} L_{\text{opt}}(\theta) \tag{10}$$

where θ denotes the model parameters and L_{opt} is the optimized loss function. By minimizing this loss function, the model can learn more accurate and generalizable feature representations based on limited few-shot data. The detailed algorithm is presented in Algorithm 2.

Algorithm 2: Few-Shot Feature Expansion Algorithm Based on Weakly Supervised Learning

```
Input: labeled sample set $\{x_1, x_2, . . . , x_n\}$, unlabeled data $\{u_1, u_2, . . . , u_m\}$, rule base
Output: expanded feature set $\{f1', f2', . . . , fn'\}$
Data preprocessing: normalize all samples
$\mu_f = \text{mean}(f_i), \sigma_f = \text{std}(f_i)$
Normalized features: $f_i' = \frac{f_i - \mu_f}{\sigma_f}$
Weak supervision signal generation: generate pseudo-labels for unlabeled data using the rule base
Feature space expansion: combine weak supervision signals with labeled data to expand feature dimensions
for each labeled sample $x_i$ do
    Extract feature $f_i$
    Expand the feature space using pseudo-labels generated by weak supervision signals
    Store the expanded feature $fi'$ into the feature set
end for
Weighted distance metric: $d_{\text{opt}}(f_{\text{cani}}, \mu_c) = \sqrt{\sum_{k=1}^{D} w_k (f_{\text{cani},k} - \mu_{c,k})^2}$
Output expanded feature set $\{f_1', f_2', \dots, f_n'\}$
```

3.3 Optimization Analysis Based on Strongly Labeled Algorithms

In traditional learning methods, strongly labeled samples are mainly used for prototype initialization and classifier training, which presents significant limitations. Conventional prototypical networks directly compute the mean vector of strongly labeled sample features as the class prototype, without considering the dynamic evolution of feature distributions. Such static prototypes fail to capture the diversity of class features and cannot leverage subsequently expanded features to refine their representations. In few-shot text classification tasks, models based on static prototypes are susceptible to semantic bias in samples, resulting in blurred decision boundaries [21].

Strict preprocessing is applied to strongly labeled samples, including data cleaning to remove invalid samples and normalization to unify data formats and scales. CNN or Transformer backbone networks are employed to extract high-dimensional discriminative features. The mean vector and distribution variance of each class are then computed to construct a class prototype library. Compared with traditional static prototypes, the introduction of variance enables prototypes to represent the distribution range of class features, providing a basis for subsequent outlier feature filtering.

Traditional methods require full-parameter fine-tuning when adapting to new tasks, which not only demands substantial computational resources but also easily leads to overfitting and reduced generalization. The meta-learning-based MAML algorithm adapts to new tasks through gradient iterations; however, repeated iterations may amplify the noise in strongly labeled samples [22].

To improve the quality and effectiveness of strongly labeled samples, this paper first performs rigorous preprocessing, including data cleaning, removal of invalid samples, and normalization. The normalization process ensures consistency in data format and scale, thereby avoiding bias caused by scale differences during training.

Assume a strongly labeled sample set $x_1, x_2, \ldots, x_n$, where each sample x_i belongs to class c with feature vector f_i. The features of each sample are first normalized to have zero mean and unit variance:

$$f_i' = \frac{f_i - \mu_f}{\sigma_f} \tag{11}$$

where f_i is the original feature vector of sample i, μ_f and σ_f denote the mean and standard deviation of the feature vectors, respectively, and fi' is the normalized feature.

In traditional methods, class prototypes are represented by the mean vector of each class. However, this approach relies solely on class means and ignores the distribution information among features. To address this issue, this paper introduces class variance to better characterize feature distributions. The prototype of class c is represented not only by the mean vector μ_c, but also by the covariance matrix Σ_c. The mean vector and covariance matrix of class c are computed as follows:

$$\mu_c = \frac{1}{n_c} \sum_{i=1}^{n_c} f_i' \tag{12}$$

$$\Sigma_c = \frac{1}{n_c} \sum_{i=1}^{n_c} (f_i' - \mu_c)(f_i' - \mu_c)^T \tag{13}$$

where n_c is the number of samples in class ccc, fi' is the normalized feature vector, μ_c is the class mean vector, and Σ_c is the class covariance matrix.

With the introduction of the covariance matrix, class prototypes can represent not only the central tendency but also the distribution range of class features, enabling better adaptation to feature diversity.

By incorporating variance information, class prototypes can characterize the distribution range of class features. During training, this information can be used for outlier detection by computing the distance between a feature and the class prototype. Suppose the distance between a feature vector f_{cani} and class prototype c is $d(f_{\text{cani}}, \mu_c)$; the deviation can be measured as follows:

$$d(f_{\text{cani}}, \mu_c) = \sqrt{\left(f_{\text{cani}}' - \mu_c\right)^T \Sigma_c^{-1} \left(f_{\text{cani}}' - \mu_c\right)} \tag{14}$$

where $\Sigma^{-}c$ 1 is the inverse of the covariance matrix of class c. By computing distances between samples and class prototypes, outliers can be identified, which may negatively affect model training. During training, both the mean vector and covariance matrix of strongly labeled samples are used as classification references. By incorporating class variance, prototypes better adapt to class diversity. During classification, an improved distance metric is used to compute distances between samples and class prototypes, and samples are assigned to the class with the minimum distance.

Through the above optimization, the proposed strongly supervised algorithm overcomes the limitations of static prototypes, improves adaptability to class diversity, effectively utilizes variance information for feature selection, reduces overfitting, and enhances generalization. The detailed procedure is shown in Algorithm 3.

Algorithm 3: Optimization Analysis Based on Strongly Labeled Data

```
Input: strongly labeled sample set $\{x_1, x_2, \ldots, x_n\}$, feature vectors $f_i$
Output: optimized class prototypes
Data preprocessing: normalize sample features
$\mu_f = \text{mean}(f_i)$, $\sigma_f = \text{std}(f_i)$
Normalized features: $f'_i = \frac{f_i - \mu_f}{\sigma_f}$
for each class c do
    Compute class mean vector: $\mu_c = \frac{1}{n_c}\sum_{i=1}^{n_c} f'_i$
    Compute class covariance matrix: $\Sigma_c = \frac{1}{n_c}\sum_{i=1}^{n_c}(f'_i - \mu_c)(f'_i - \mu_c)^T$
end for
Compute distance between each sample and class prototype: $d(f_{\text{cani}}, \mu_c) = \sqrt{(f'_{\text{cani}} - \mu_c)^T \Sigma_c^{-1} (f'_{\text{cani}} - \mu_c)}$
Outlier filtering: if $d(f_{\text{cani}}, \mu_c) >$ threshold, discard the sample
Output optimized class prototypes $\mu_c$, $\Sigma_c$
```

4. Simulation Experiments

4.1 Simulation Environment

In this study, the design and implementation of the simulation environment are critical for evaluating the performance of different algorithms. The simulation setup, platform, parameter configuration, datasets, and evaluation metrics are described as follows.

- Hardware platform: A high-performance computing server equipped with multiple processing cores and large-capacity memory is used to handle large-scale datasets. The configuration is as follows:
  - CPU: Intel Xeon Gold 6248 (20 cores, 40 threads)
  - GPU: NVIDIA Tesla V100 (16 GB)
  - Memory: 128 GB
  - Storage: 1 TB SSD for fast data read/write
- Software platform and tools:
  - Programming environment: Python 3.8
  - Deep learning frameworks: TensorFlow 2.0, PyTorch 1.7
  - Data processing libraries: NumPy, Pandas, Scikit-learn
  - Visualization tools: Matplotlib, Seaborn
  - Optimization library: Optuna

To evaluate the performance advantages of the proposed methods, experiments are conducted on standard public datasets (CIFAR-10 [23] and MNIST [24]). The results are shown in Table 1. The few-shot feature expansion algorithm based on weakly supervised learning and the optimized strongly supervised algorithm outperform traditional methods in terms of F1-score and recall, especially in class-imbalanced tasks. Specifically, the weakly supervised learning algorithm effectively addresses class imbalance through feature space expansion and weighted distance metrics, achieving significant improvements across multiple experiments.

4.2 Evaluation Metrics

In machine learning and deep learning tasks, evaluating model performance is crucial. The confusion matrix is a visualization tool used to show the relationship between model predictions and true labels. The confusion matrix consists of four key elements: TP (True Positive): the number of positive samples correctly predicted as positive by the model. TN (True Negative): the number of negative samples correctly predicted as negative. FP (False Positive): the number of negative samples incorrectly predicted as positive. FN (False Negative): the number of positive samples incorrectly predicted as negative.

The confusion matrix provides an intuitive view of a model's classification performance, especially when classes are imbalanced. Depending on the task and objectives, different evaluation metrics can be derived from it. For classification tasks, evaluation metrics not only indicate overall model performance but also reveal the model's ability to distinguish between different classes. The main evaluation metrics used in this study are as follows:

Accuracy is one of the most commonly used metrics, measuring the proportion of correctly predicted samples over the total number of samples. It is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

where TP (True Positive) denotes the number of samples correctly predicted as positive, TN (True Negative) denotes the number of samples correctly predicted as negative, FP (False Positive) denotes the number of negative samples incorrectly predicted as positive, and FN (False Negative) denotes the number of positive samples incorrectly predicted as negative.

Accuracy is suitable for scenarios with balanced class distributions; however, in imbalanced settings, it may not fully reflect model performance.

Precision measures the proportion of correctly predicted positive samples among all samples predicted as positive. High precision indicates reliable positive predictions. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

Recall measures the proportion of positive samples correctly identified by the model. High recall indicates strong recognition capability for positive samples. It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

F1-score is the harmonic mean of precision and recall, providing a balanced measure, especially useful in class-imbalanced scenarios. It can serve as a more representative evaluation metric when class distributions are uneven. It is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

The F1-score balances precision and recall and is particularly suitable when class distributions are imbalanced.

The AUC-ROC curve is another widely used metric for evaluating classification performance. The ROC curve plots the relationship between the False Positive Rate (FPR) and the True Positive Rate (TPR), where the false positive rate is defined as:

$$\text{FPR} = \frac{FP}{FP + TN} \tag{19}$$

The true positive rate (TPR), which is equivalent to recall, is calculated as:

$$\text{TPR} = \frac{TP}{TP + FN} \tag{20}$$

The AUC value (Area Under the Curve) reflects model performance across all possible thresholds; a higher AUC indicates better model performance. The AUC-ROC curve is applicable to multi-class classification tasks and helps assess model performance under different decision thresholds.

4.3 Performance Comparison and Analysis

This study compares the algorithms from references [25-27]. Reference [25] proposed an SVM optimization algorithm under weakly supervised settings, incorporating weak labels to improve performance in few-shot learning. Reference [26] proposed a pseudo-label-based semi-supervised learning algorithm for few-shot modulation recognition tasks. This approach improves classification performance in few-shot settings via pseudo-label generation and self-learning mechanisms. Reference [27] introduced a few-shot remote sensing image object detection algorithm based on meta-learning. By training the model within a meta-learning framework, it can rapidly adapt to new target classes and achieves good performance in remote sensing classification tasks.

Table 1: Experimental comparison of different algorithms

Algorithm	Accuracy	Precision	Recall	F1-score
Weakly Supervised Few-Shot Feature Expansion Algorithm	85%	87%	84%	85.5%
Strongly Labeled Algorithm	88%	89%	87%	88%
Ref. [25]	83%	84%	82%	83%
Ref. [26]	80%	81%	79%	80%
Ref. [27]	84%	85%	83%	84%

As shown in Table 1, the weakly supervised few-shot feature expansion algorithm improves over reference [25] by 2.35% in accuracy, 3.45% in precision, 2.38% in recall, and 2.93% in F1-score. Compared to reference [26], improvements are 5.88%, 6.90%, 5.95%, and 6.50%, respectively. Compared to reference [27], improvements are 1.18%, 2.30%, 1.19%, and 1.76%, respectively. For the strong-label algorithm, compared with Reference [25], the accuracy increased by 5.68%, precision increased by 5.62%, recall increased by 5.75%, and the F1 score increased by 5.68%. Compared with Reference [26], the accuracy increased by 9.09%, precision increased by 8.99%, recall increased by 9.20%, and the F1 score increased by 9.09%. Compared with Reference [27], the accuracy increased by 4.55%, precision increased by 4.49%, recall increased by 4.60%, and the F1 score increased by 4.55%. The improvements of the few-shot feature expansion algorithm based on weakly supervised learning over other algorithms can be primarily attributed to its ability to expand the feature space by incorporating unlabeled data and pseudo-labels, thereby enhancing the model's discriminative capability for few-shot classes. This algorithm effectively leverages weak supervision signals, not only compensating for the scarcity of labeled data but also improving class separability through feature expansion, demonstrating particularly strong performance in scenarios with class imbalance and limited samples.

Compared with the method in [25], the advantage of the few-shot feature expansion algorithm based on weakly supervised learning lies in its ability to expand the feature space, thereby reducing the model's reliance on a single class mean vector and avoiding the overfitting issues common in traditional approaches, which ultimately improves overall classification performance. In comparison with [26], the superior performance of this algorithm is attributed to its use of unlabeled data to generate pseudo-labels, augmenting the training data and effectively reducing bias while enhancing model robustness, particularly under class imbalance conditions. The C2 algorithm, in contrast, fails to fully exploit additional unlabeled data when dealing with scarce samples and imbalanced classes, and thus does not achieve comparable improvements in classification performance. Relative to [27], the improvements of the weakly supervised few-shot feature expansion algorithm stem from its more flexible feature expansion methodology, especially through weighted feature dimensions, which helps maintain strong classification performance even under complex class distributions. Although the C3 algorithm can adapt to new classes through meta-learning, its capabilities for feature expansion and class discrimination are relatively limited, resulting in slightly inferior performance on complex tasks compared with our algorithm. The performance improvements of the strong-label algorithm are mainly due to the introduction of class variance information, which allows class prototypes to represent not only the mean vector but also the full distribution of class features through the variance matrix. This optimization enables the strong-label algorithm to effectively reduce misclassifications and enhance generalization when facing imbalanced samples and diverse class characteristics. Compared with [25], the improvements in the strong-label algorithm lie in the incorporation of class variance and weighted features, allowing the algorithm to accurately represent the class center while also capturing the distribution range of the class, thereby enhancing the model's adaptability to complex features. In comparison with [26], the advantage of the strong-label algorithm is that, by introducing variance information and optimizing inter-class distances, it better handles class imbalance and improves both feature selection and class discrimination capabilities. The C2 algorithm, however, does not fully utilize class distribution information, resulting in weaker performance on these tasks.

Compared with [27], the strong-label algorithm further optimizes inter-class feature differentiation through the variance matrix and weighted feature expansion, enhancing discriminability between classes, particularly when class feature distributions are complex, which significantly improves classification accuracy. While the C3 algorithm can adapt to new classes, its handling of class differences is relatively simplistic, and therefore it cannot match the advantages of the strong-label algorithm.

Table 2: Comparison of computational complexity, training time, robustness, and sensitivity to outliers among different algorithms

Algorithm	Computation Complexity	Training Time	Robustness	Sensitivity to Outliers
Weakly Supervised Few-Shot Feature Expansion Algorithm	Medium	relatively long	High	Medium
Strongly Labeled Algorithm	High	relatively long	High	Low
Ref. [25]	Low	Short	Low	High
Ref. [26]	Medium	Medium	Medium	High
Ref. [27]	High	Long	High	Medium

As shown in Table 2, first, both the Weakly Supervised Few-Shot Feature Expansion algorithm and the Strong-Label Algorithm exhibit relatively high computational complexity, especially the Strong-Label Algorithm. By incorporating class variance information and weighted feature optimization, it imposes an additional computational burden, making it suitable for scenarios that require substantial computing resources. However, these algorithms are able to achieve significant performance improvements in class-imbalanced and few-shot learning tasks by effectively expanding the feature space, introducing pseudo-labels, and optimizing feature selection. In contrast, the algorithm in [25] has lower computational complexity and is suitable for simpler tasks, but it cannot adequately handle large-scale datasets or complex scenarios, limiting its overall performance. The algorithm in [26] has moderate computational complexity, but its computational efficiency is lower than that of the Strong-Label Algorithm and the Weakly Supervised Algorithm when handling complex tasks. Regarding training time, the Weakly Supervised Few-Shot Feature Expansion algorithm and the Strong-Label Algorithm require relatively long training periods, particularly when processing large datasets, as the additional computational load from feature expansion increases training time. Nonetheless, through parallel computing and optimized training pipelines, these algorithms maintain relatively high efficiency. In comparison, the algorithms in [25] and [26] have shorter training times, suitable for quick experiments on smaller datasets, but they may encounter bottlenecks when handling large-scale data. The algorithm in [27], which leverages a meta-learning framework to adapt to new tasks, also demonstrates strong performance; however, multiple iterations are required, leading to longer training times, particularly when the number of tasks increases. In terms of robustness, the Weakly Supervised Few-Shot Feature Expansion algorithm and the Strong-Label Algorithm demonstrate strong robustness, particularly when faced with class imbalance and scarce sample problems. By combining feature expansion with weakly supervised signals, these algorithms enhance model adaptability and stability. In contrast, the algorithm in [25] exhibits poor robustness, especially under class imbalance, as it relies on simple Euclidean distance calculations and cannot effectively handle complex class distributions. The algorithm in [26], although enhanced through pseudo-label generation, still shows limited robustness under class imbalance or uneven data distributions, which may lead to performance degradation. While the algorithm in [27] demonstrates strong adaptability through meta-learning, it is sensitive to noise and outliers in the data, resulting in relatively lower robustness in complex scenarios. Regarding sensitivity to outliers, both the Weakly Supervised Few-Shot Feature Expansion algorithm and the Strong-Label Algorithm show strong resistance to interference. The Strong-Label Algorithm, by incorporating class variance and weighted features, maintains stability in the presence of outliers, effectively mitigating their negative impact on the model. In contrast, the algorithms in [25] and [26] are highly sensitive to outliers, especially when the dataset contains substantial noise, which significantly affects their performance. Although the meta-learning framework in [27] provides strong adaptability, the multiple iterations and training requirements make it more sensitive to outliers, potentially affecting performance on noisy datasets.

Overall, the Weakly Supervised Few-Shot Feature Expansion algorithm and the Strong-Label Algorithm demonstrate clear advantages when dealing with few-shot learning tasks, class imbalance, and scarce data, particularly in terms of robustness and adaptability to outliers, enhancing both classification accuracy and stability. Traditional algorithms [25] and [26] perform well in simple tasks but are limited in complex tasks and noisy environments. The algorithm in [27], although adaptive, faces challenges in practical applications due to its computational complexity and sensitivity to outliers. These comparative results provide deeper insights, indicating that algorithms based on weak supervision and strong labels have stronger practical potential in few-shot learning scenarios.

5. Conclusion

This paper proposes an algorithm that integrates few-shot learning with weakly supervised learning and conducts simulation experiments to evaluate its performance. The core design of the algorithm includes a semantic–syntactic matching rule system, which generates pseudo-labels for massive unlabeled data, transforming them into weak supervision signals for model training. This approach greatly improves the utilization efficiency of large-scale unlabeled datasets. Additionally, a weighted distance metric is introduced to assign different weights to different feature dimensions, emphasizing key features while reducing the interference of redundant features on the model. A variance matrix is also employed to identify outlier samples, mitigating the impact of noise on model training and enhancing classification accuracy and stability. The simulation experiments use the standard public datasets CIFAR-10 and MNIST. On these datasets, both the Weakly Supervised Few-Shot Feature Expansion algorithm and the Strong-Label Algorithm achieve improvements in accuracy, precision, recall, and F1 score. Specifically, the Weakly Supervised Few-Shot Feature Expansion algorithm achieves a 2.35% increase in accuracy, a 3.45% increase in precision, a 2.38% increase in recall, and a 2.93% increase in F1 score compared to [25]. Compared to [26], accuracy increases by 5.88%, precision by 6.90%, recall by 5.95%, and F1 score by 6.50%. Compared to [27], accuracy increases by 1.18%, precision by 2.30%, recall by 1.19%, and F1 score by 1.76%. For the Strong-Label Algorithm, accuracy, precision, recall, and F1 score improve by 5.68%, 5.62%, 5.75%, and 5.68%, respectively, compared to [25]. Compared to [26], accuracy increases by 9.09%, precision by 8.99%, recall by 9.20%, and F1 score by 9.09%. Compared to [27], accuracy increases by 4.55%, precision by 4.49%, recall by 4.60%, and F1 score by 4.55%. Compared with [25], the Weakly Supervised Few-Shot Feature Expansion algorithm reduces the model's reliance on single-class mean vectors, thereby improving classification capability, while the Strong-Label Algorithm introduces class variance and weighted features, enhancing its ability to adapt to complex feature distributions. To address the poor model performance under class imbalance reported in [26], this study generates pseudo-labels and applies them in model training, effectively improving classification performance. Furthermore, by incorporating variance information and inter-class distance optimization, feature selection and discriminative capability are enhanced. Compared with [27], the Weakly Supervised Few-Shot Feature Expansion algorithm demonstrates more flexible weighting, resulting in superior classification performance, while the Strong-Label Algorithm further improves classification accuracy under complex feature distributions through variance matrices and weighted feature expansion. The results of the simulation experiments provide valuable reference data for model training in data-scarce scenarios.

There are still some limitations in the current research. The quality of pseudo-labels depends on the completeness of the initial rule base; in semantically complex scenarios with an insufficient initial library, training performance may decrease. Moreover, the simulation experiments only use standard public datasets, and no testing has been conducted on other complex datasets, so the algorithm's adaptability requires further validation. In future research, the proposed algorithm could be combined with artificial intelligence and reinforcement learning techniques to extend it from static environments to dynamic and adaptive scenarios, enhancing its automation and intelligence value. Additionally, the algorithm could be integrated with pseudo-label quality enhancement strategies to enable the application and analysis of self-supervised learning within this framework.

References

[1] Ren, D., Wang, Q., Wei, Y., Meng, D., & Zuo, W. (2022). Research progress on visual weakly supervised learning. *Journal of Image and Graphics*, 27(6), 1768–1798.

[2] Tian, X., Wang, L., & Ding, Q. (2019). A survey of image semantic segmentation methods based on deep learning. *Journal of Software*, 30(2), 440–468. https://doi.org/10.13328/j.cnki.jos.005659

[3] Liu, Y., Shao, M., Zhang, L., & Shao, J. (2025). Prototype optimization method for few-shot learning based on semantics. *Pattern Recognition and Artificial Intelligence*, 38(2), 132–142. https://doi.org/10.16451/j.cnki.issn1003-6059.202502003

[4] He, X., & Lin, J. (2022). Fine-grained few-shot learning with weakly supervised object localization. *Journal of Image and Graphics*, 27(7), 2226–2239.

[5] Zhao, K., Jin, X., & Wang, Y. (2021). A survey on few-shot learning. *Journal of Software*, 32(2), 349–369. https://doi.org/10.13328/j.cnki.jos.006138

[6] Pan, C., Huang, J., Hao, J., Gong, J., & Zhang, Z. (2020). A survey of weakly supervised learning methods combining zero-shot learning and few-shot learning. *Systems Engineering and Electronics*, 42(10), 2246–2256.

[7] Li, R., Wei, Z., Fan, Y., Ye, S., & Zhang, G. (2024). Few-shot text classification method with enhanced prompt learning. *Acta Scientiarum Naturalium Universitatis Pekinensis*, 60(1), 1–12. https://doi.org/10.13209/j.0479-8023.2023.071

[8] Liu, Y., Lei, Y., Fan, J., Wang, F., Gong, Y., & Tian, Q. (2021). A survey on image classification techniques based on few-shot learning. *Acta Automatica Sinica*, 47(2), 297–315. https://doi.org/10.16383/j.aas.c190720

[9] Zhu, J., Yao, G., Zhang, G., Li, J., Yang, Q., Wang, S., & Ye, S. (2021). A survey on few-shot learning with deep neural networks. *Computer Engineering and Applications*, 57(7), 22–33.

[10] Zheng, W., Fu, S., Chen, J., Peng, Q., Tu, Y., Zou, B., & You, X. (2025). Few-shot graph anomaly detection under extremely weak supervision. *Chinese Journal of Computers*, 48(4), 927–948.

[11] Li, Y., Xu, C., Tang, X., & Li, X. (2023). A survey of semi-supervised learning methods. *World Science and Technology Research and Development*, 45(1), 26–40. https://doi.org/10.16507/j.issn.1006-6055.2022.07.001

[12] Yang, H., Quan, J., Liang, X., & Wang, Z. (2021). Research progress on object detection based on weakly supervised learning. *Computer Engineering and Applications*, 57(16), 40–49.

[13] Xu, D., & Wu, Y. (2024). Research progress on deep learning algorithms for object detection in optical remote sensing images. *Journal of Remote Sensing*, 28(12), 3045–3073.

[14] Li, Y., Wang, P., Liu, Y., Liu, G., Wang, C., Liu, X., & Guo, M. (2020). Weakly supervised real-time object detection based on saliency maps. *Acta Automatica Sinica*, 46(2), 242–255. https://doi.org/10.16383/j.aas.c180789

[15] Zhou, M., & Wang, X. (2018). Weakly supervised deep neural network model for remote sensing image object detection. *Science China Information Sciences*, 48(8), 1022–1034.

[16] Sun, M., Lü, C., Han, Y., Li, S., & Wang, Z. (2021). Surface defect detection based on weakly supervised learning with attention mechanism. *Journal of Computer-Aided Design & Computer Graphics*, 33(6), 920–928.

[17] Li, Z., Jia, L., Zhang, B., & Li, P. (2025). Few-shot image classification based on self-supervised learning and second-order representation. *Chinese Journal of Computers*, 48(3), 586–601.

[18] Shi, Y., Shi, D., Qiao, Z., Zhang, Y., Liu, Y., & Yang, S. (2023). A survey on few-shot object detection. *Chinese Journal of Computers*, 46(8), 1753–1780.

[19] An, S., Guo, Y., Bai, Y., & Wang, T. (2023). A survey on few-shot image classification. *Computer Science and Exploration*, 17(3), 511–532.

[20] Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30.

[21] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning* (pp. 1126–1135).

[22] Abouelnaga, Y., Ali, O. S., Rady, H., & Moustafa, M. (2016). CIFAR-10: KNN-based ensemble of classifiers. *IEEE*.

[23] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms.

[24] Ding, S., Sun, Y., Liang, Z., Guo, L., Zhang, J., & Xu, X. (2024). A survey on support vector machine algorithms under weak supervision. *Chinese Journal of Computers*, 47(5), 987–1009.

[25] Shi, Y., Xu, H., & Liu, Y. (2020). A few-shot modulation recognition algorithm based on pseudo-label semi-supervised learning. *Journal of Northwestern Polytechnical University*, 38(5), 1074–1083.

[26] Li, H., Wang, Y., & Yang, L. (2024). Few-shot object detection in remote sensing images based on meta-learning. *Journal of Beijing University of Aeronautics and Astronautics*, 50(8), 2503–2513. https://doi.org/10.13700/j.bh.1001-5965.2022.0637

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Copyrights