A knowledge distillation-based multi-scale relation-prototypical network for cross-domain few-shot defect classification

A Knowledge Distillation-Based Multi-Scale Relation-Prototypical Network for Cross-Domain Few-Shot Defect Classification

Jiaqi Zhao[1†], Xiaolong Qian[1*], Yunzhou Zhang[1], Dexing Shan[1], Xiaozheng Liu[1], Sonya Coleman[2] and Dermot Kerr[2]

[1*]College of Information Science and Engineering, Northeastern University, Shenyang, China.
[2]Intelligent Systems Research Centre, University of Ulster, Londonderry, U.K..

*Corresponding author(s). E-mail(s): qianxiaolong@ise.neu.edu.cn;
Contributing authors: 2000925@stu.neu.edu.cn;
†These authors contributed equally to this work.

Abstract

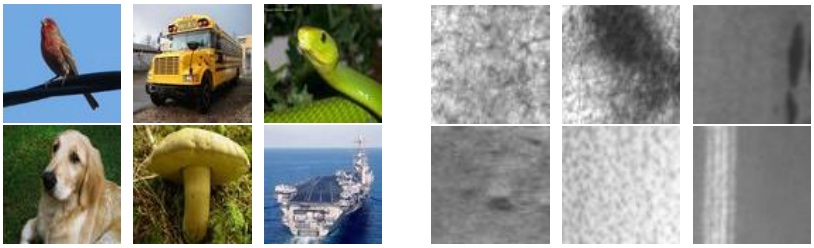
Surface defect classification plays a very important role in industrial production and mechanical manufacturing. However, there are currently some challenges hindering its use. The first is the similarity of different defect samples makes classification a difficult task. Second, the lack of defect samples leads to poor accuracies when using deep learning methods. In this paper, we first design a novel backbone network, ResMSNet, which draws on the idea of multi-scale feature extraction for small discriminative regions in defect samples. Then, we introduce few-shot learning for defect classification and propose a Relation-Prototypical network (RPNet), which combines the characteristics of ProtoNet and RelationNet and provides classification by linking the prototypes distances and the nonlinear relation scores. Next, we consider a more realistic scenario where the base dataset for training the model and target defect dataset for applying the model are usually obtained from domains with large differences, called cross-domain few-shot learning (CD-FSL). Hence, we further improve RPNet to KD-RPNet inspired by knowledge distillation methods. Through extensive comparative experiments and ablation experiments, we demonstrate that either

our ResMSNet or RPNet proves its effectiveness and KD-RPNet outperforms other state-of-the-art approaches for few-shot defect classification.

Keywords: few-shot learning, Defect classification, Multi-scale feature encoder, Cross-domain, Knowledge distillation

1 Introduction

The vigorous development of industry and manufacturing is beneficial to economic growth and daily life. However, due to unexpected factors in the production process, industrial products may to subject to defects, either on the surface or within the product. The defects may not only affect the use of products or bring economic losses, but also cause serious safety issues and accidents. Therefore, surface defect recognition has become an urgent problem to be solved.



(a) (b)

Fig. 1. Comparison of (a) *mini*Imagenet dataset and (b) NEU-CLS defect dataset.

There are many approaches to surface defect recognition, from manual detection methods to computer vision detection methods. In recent years, with the wide application of deep learning technology, researchers have designed lots of effective networks based on convolutional neural networks (CNNs) for defect classification and achieved very good results. However, there remains two problems. First, compared with samples in common image datasets, the similarity among different defect samples is high in defect detection datasets, as shown in Fig. 1, and therefore it is difficult to predict categories. Second, approaches based on deep learning require large-scale labeled datasets for training, but the generation of defect samples belongs to small probability events in some scenarios, so it is difficult to collect enough defect samples for adequate training. Hence the accuracy of deep learning methods decreases significantly.

To address the first problem of the high similarity of defect samples, a learning network is required to identify small and critical discriminative regions

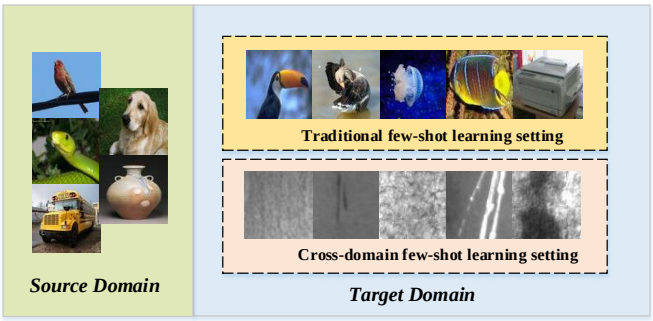


Fig. 2. Traditional and cross-domain few-shot learning setting

in defect images to predict the correct category. However, it is difficult to achieve this for common feature encoders such as VGG and ResNet. Therefore, we propose to learn from the idea of multi-scale feature extraction commonly used in the research of small object detection, so that the model can pay more attention to the tiny and discriminative regions features. Inspired by [1], we propose a novel multi-scale feature encoder called ResMSNet where the bottleneck structure in ResNet is replaced with a ResMSNet block with multiple residual-like structures to expand the receptive field, and the output of different combinations of scaled receptive fields improves the representation ability of local features of defect samples.

To solve the second issue, we propose to adopt few-shot learning methods as traditional convolutional neural networks require large-scale labeled dataset which are not available for the task of surface defect detection. Few-shot learning is an emerging research direction in artificial intelligence in recent years. The purpose is to enable the network to recognize an unknown object using very few samples. Most existing few-shot learning methods are usually divided into a meta-training stage and a meta-testing stage. First of all, it is necessary to prepare a labeled base dataset and a disjoint target set, and both are split into support sets and query sets during training and testing. At the meta-training stage, a base dataset is used to train the model such that the model acquires the few-shot learning ability. Then the model will be evaluated on the query sets of target dataset after being adapted with support sets at the meta-testing stage. These existing few-shot classification methods perform successfully, however, the base dataset and the target dataset are usually obtained from the same large-scale dataset, while in many real scenarios is difficult to obtain base sets and target sets from the same scenario. The target domain may be quite different from the source domain which can also be considered a large domain shift, such as the source domain is natural images but the target domain is visual inspection images. In such an example, the performance of the methods falls sharply under traditional few-shot learning settings. To distinguish from traditional few-shot learning, the scenario when the source and target domains are different is referred to as cross-domain

few-shot learning (CD-FSL). Fig. 2 illustrates the two settings. Unfortunately, there is little research on few-shot defect classification, let alone more suitable cross-domain few-shot defect classification methods. It is worth mentioning that although there are also studies on zero-shot learning, in general the performance of zero-shot methods is not as good as few-shot methods, and when the domain shift is large, the performance will be much worse, so thus it is not suitable for defect classification tasks.

For traditional few shot learning, we propose a Relation-Prototypical Network (RPNet). RPNet contains a prototypical branch and a relation branch. We use the relation scores output by the relation branch to correct the prototypical distances output by the prototypical branch, so that the distance between the prototypes of the same class is close, and the heterogeneity is further away, thereby increasing the classification accuracy. Extensive experimental results show that our RPNet outperforms few-shot learning baselines on NEU-CLS defect dataset and demonstrates an advantage compared with existing methods on common datasets. For CD-FSL, we employ a knowledge distillation based approach and improve RPNet to KD-RPNet, an end-to-end knowledge distillation based RPNet. We first collect and devise a new few-shot ventilation pipeline inner surface defect dataset, namely Pipe-Defect. We denote it as a target dataset as well as NEU-CLS. Then we respectively train the student RPNet with labeled source domain data and strongly-augmented unlabeled target domain defect data and the teacher RPNet with weakly-augmented unlabeled defect data. The comparative experimental results indicate that our KD-RPNet outperforms most existing CD-FSL methods and reaches state-of-the-art performance.

In summary, the contributions of our paper are itemized as follows:

1. A novel multi-scale backbone network, ResMSNet, is proposed to extract defect features and the performance is better than ResNet12 using the NEU-CLS dataset.
2. To solve the problem of insufficient defect samples, we introduce a Relation-Prototypical Network (RPNet) based on few-shot learning, which outperforms baselines on NEU-CLS with either ResNet12 or ResMSNet and shows advantages on common datasets compared with current methods.
3. We create a novel few-shot defect dataset called Pipe-Defect for evaluating the proposed methods.
4. For CD-FSL, we further improve RPNet based on knowledge distillation and we refer to it as KD-RPNet. After extensively evaluating performance on both NEU-CLS and Pipe-defect, we find that our approach significantly outperforms state-of-the-art CD-FSL methods.

2 Related works

2.1 Defect classification

Defect classification is an important part of surface defect recognition. In the last century, traditional defect classification mainly depends on a manual process, which inherently has issues such as high work intensity, low efficiency, high cost, poor accuracy etc. [2]. Many of these issues have been overcome with the use of computer vision for defect classification. In [3] Top Hat operators were developed with different morphology. By extracting different types of defect regions, the operator obtains characteristic parameters, establishes defect templates, and classifies different defects. In [4], binarization and morphological operations are used to predict the categories of defects. In recent years, inspired by deep learning models for object detection, some research focused on the use of deep convolution neural networks to solve defect classification problems. The work in [5] designed an end-to-end method based on CNN for steel surface defect classification, the parameters of which are randomly initialized and trained from scratch. In [6] a weakly supervised learning method is proposed known as a classification aware defect detection network (CADN). Similarly, [7] presented a segmentation-based deep-learning architecture which was designed for surface defect detection and segmentation. The model was trained on pixel-wise labels of the defect and a decision network was built to predict the the existence of defects in the whole image. In [8] a bioinspired visual-integrated model (BIVI-ML) was introduced, where a visual attention mechanism is designed to reduce the inference of the complex texture background. The main challenge of applying such methods to defect recognition tasks is that different products often generate different types of defect. Therefore, it is necessary to use the product specific dataset to train the model. However, the generation of defects belongs to small probability events, which makes it difficult to collect enough defect samples. This hinders the application of artificial intelligence technology to defect classification.

2.2 Few-shot learning

Few-shot learning [9–11] is a new research topic in recent years. The purpose is to solve the problem of insufficient training samples in machine learning tasks. Few-shot learning methods can be summarized as meta-learning methods, transfer learning methods [12, 13] and semi-supervised methods [14–16], among which meta-learning methods are the mainstream method. MatchingNet [17] adopts the mechanism of attention and external memory, and compares the cosine distance between the support features of each class and the query features. ProtoNet [18] learns a metric space and compares the Euclidean distances of query prototypes and support prototypes in this space. RelationNet [19] describes a learnable non-linear comparator to replace the traditional distance based linear comparator to judge the relationship between query and support features. MAML [20] is an optimization-based meta-learning

method. Its purpose is to learn appropriate initialization parameters to enable it to quickly adapt to new tasks through few samples. In the field of defect recognition, the few-shot learning method is still in its infancy. TGRNet [21] introduced a general few-shot surface defect segmentation theory for metals, and a novel multi-graph reasoning module is proposed for few-shot semantic segmentation tasks by exploring the similarities between images. In short, few-shot defect classification approaches are of great research value.

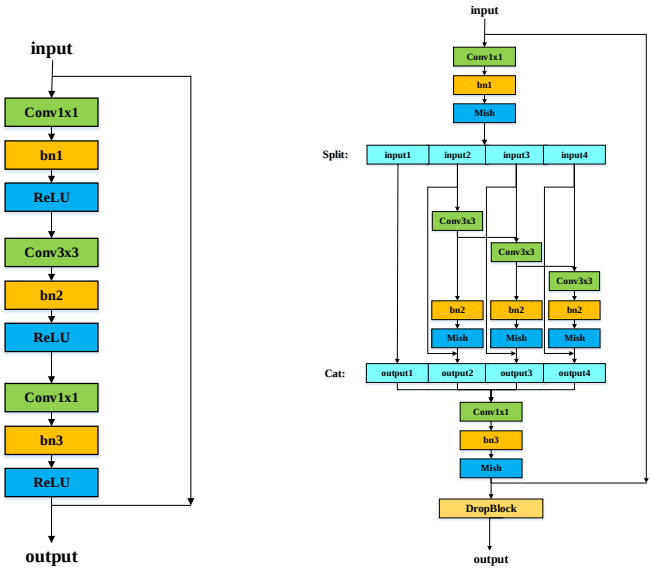
2.3 Cross-domain few-shot learning

Cross-domain few-shot learning (CD-FSL) is a realistic setting for evaluation where base and novel classes are sampled from different domains. The work in [22] found that traditional few-shot classification methods fail to address such domain shifts and even worse than the baseline method. To improve the accuracy when domain shifts exist, [23] proposed a learnable feature-wise transformation layer used in a feature encoder which is able to reduce the distance between different domains. The method is the first approach to address the CD-FSL issue but the gap between domains is small. In [24] a novel Broader Study of Cross-Domain Few-Shot Learning benchmark (BSCD-FSL), consisting of images from multiple image types that differ from natural images is established and indicated that meta-learning based few-shot learning methods performed worse than simple fine-tuning methods. To deal with cross-domain problems in few-shot defect classification tasks, a novel attention and adaptive bilinear matching network called AABM [25] is introduced. Similarly, [26] a graph embedding and distribution transformation (GEDT) module and a optimal transport (OPT) module to discover more potentially useful information between samples and classify the test samples by minimizing Wasserstein distance. In addition, recent works have utilized unlabeled data from the target domain to learn specific representations and achieved good performance. A dynamic distillation based approach is designed in [27] and used to train a feature extractor with augmented unlabeled target data and labeled source data so that it could evaluate few-shot learning performance on the target domain. Our KD-RPNet is a similar approach to this, however, is end-to-end.

3 Proposed methods

3.1 Problem definition

In few-shot learning, we use a N-way K-shot setting, such that images of support sets are collected from N classes, and each class contains K images. Similar to general deep learning methods, the dataset needs to be divided into a training set and a testing set. During an episode in the training stage, the algorithm will randomly sample N categories and construct a few-shot learning classification task. The task consists of a support set and a query set, which share the label space, but when testing, the label space is disjoint.



(a) (b)

Fig. 3. Comparison between the bottleneck block and ResMSNet block

For RPNet using the few-shot learning settings, we define the ResMSNet feature encoder as f_r, a labeled support sample set as $S = \{\mathbf{x}_i\}_{i=1}^{M}$, a query set as $Q = \{\mathbf{x}_j\}_{j=1}^{N}$, the corresponding categorical labels as y_i and y_j and S_k denotes the labeled support samples set belongs to class k.

For KD-RPNet using the CD-FSL setting, we name the labeled data from source domain $D_S = \{(\mathbf{x}_i^S, y_i^S)\}_{i=1}^{M_S}$ and the unlabeled data $D_T = \{\mathbf{x}_j^T\}_{j=1}^{N_T}$ from target domain. We further define the teacher RPNet as g_t and the student RPNet as g_s.

3.2 ResMSNet

In this subsection, we will describe ResMSNet, a novel multi-scale feature encoder for defect features extraction.

ResMSNet is an improvement and promotion of the ResNet12 feature encoder commonly used in few-shot learning methods. The difference between the two is that the ResMSNet block, which gives the ability of multi-scale defect feature extraction, replaces the traditional bottleneck block.

Fig. 3 shows a comparison of the bottleneck block and ResMSNet block. Fig. 3a is the bottleneck block of the classical ResNet network, which is composed of a 1×1, a 3×3, and another 1×1 convolution layer. The first 1×1 layer is responsible for reducing dimensions and the other one is for increasing, such that the number of parameters in the overall calculation is reduced compared with the basic block. Although the ResNet network constructed using the bottleneck block is excellent, its performance is still insufficient for defect

samples with high similarity and small defect resolution. Taking these factors into account, we decide to use a multi-scale feature encoder to extract defect features. However, traditional multi-scale feature encoders such as Feature Pyramid Network (FPN) have complex structures and require high computational resources. It is worth mentioning that Res2Net inspires us to increase the receptive field of the model by modifying the structure of the block, rather than merging on the basis of layers, thus we design a ResMSNet block accordingly to improve ResNet12.

Fig. 3b shows the ResMSNet block. Here, a smaller set of 3×3 convolution kernels is used to replace the 3×3 convolution kernel in the bottleneck block, and each filter group is linked with a similar residual-like structure. This change enables each 3×3 convolution operation to potentially accept the previous set of feature information, and each output can increase the receptive field. Hence each block can obtain feature combinations with different numbers and sizes of receptive fields. Moreover, we continue to add the residual-like structure inside each filter group, and connect the feature maps before and after convolution in the same group. The calculation process is as follows: we input the defect image into a 1×1 convolution kernel, and then split the obtained feature map into s sub-feature maps. Each sub-feature map is denoted as $\mathbf{u}_i$, where $i \in 1, 2, ..., s$. The number of channels of each sub-feature map is $1/s$ of the input feature map while the scale size is the same as the input feature map. In order to reduce the number of parameters, we use a piecewise function such that: when i=1, the output $\mathbf{v}_i$ is the same as the input sub-feature map $\mathbf{u}_i$; when i=2, in addition to $\mathbf{u}_i$ sub-feature map, each sub-feature map contains a small 3×3 convolution kernel, denoted as $\boldsymbol{C}_i(,)$ which is convolved with $\mathbf{u}_i$; and in all other cases $\mathbf{u}_i$ is added to the output feature $\mathbf{v}_{i-1}$, then fed into $\boldsymbol{C}_i(,)$, and the convolution result obtained is then added to the sub-feature map $\mathbf{u}_i$. The whole calculation process can be written as:

$$\mathbf{v}_i = \begin{cases} \mathbf{u}_i, & i = 1; \\ \boldsymbol{C}_i(\mathbf{u}_i) + \mathbf{u}_i, & i = 2; \\ \boldsymbol{C}_i(\mathbf{u}_i + \mathbf{v}_{i-1}) + \mathbf{u}_i, & 2 < i \leq s. \end{cases} \tag{1}$$

At the end of this series of convolutions, all output sub-feature maps are concatenated and passed to a 1×1 convolution to obtain the output feature map fused with more detailed feature information. In addition, we also replace the activation function with Mish [28] from ReLU. Mish is a self-regularized non-monotonic neural activation function. The smooth activation function provides better information as inputs to the neural network, thus ensuring better performance and improved generalization ability. The Mish activation function is denoted as:

$$\text{Mish}(x) = x \times \tanh(\ln(1 + e^x)) \tag{2}$$

where x is a variable that input to the function.

Furthermore, we use the DropBlock [29] to avoid overfitting, which is a structured and two-dimensional Dropout. This method involves dropping

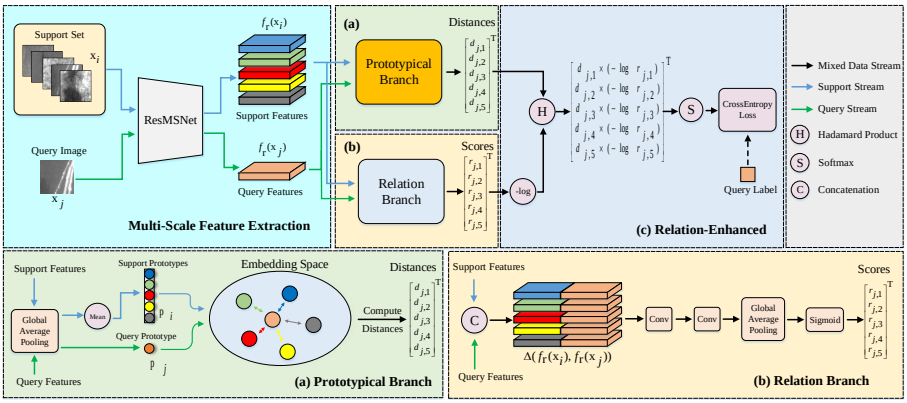


Fig. 4. Relation-Prototypical Network (RPNet) for few-shot defect classification (5-way 1-shot). (a) shows the prototypical branch, which compute distances between support and query prototypes in the embedding space. (b) shows the relation branch, whose aim is to judge the similarity by calculating the relation scores. (c) shows the relation-enhanced module , it is the key to improve the performance of RPNet by using relation scores to adjust the prototype distances

the adjacent regions of the layer feature map instead of dropping the individual random elements can effectively improve the accuracy and robustness of the algorithm. Finally, we replace all the bottleneck blocks in ResNet12 with ResMSNet blocks, to obtain a new feature encoder named ResMSNet backbone.

In our approach, we use s as the control parameter of the scale dimension, where a larger s gives the backbone a larger receptive field. Specifically, in our experiment using the NEU-CLS dataset, we imperially determined that when $s = 3$, the best accuracy is be obtained. The best s may be different on different datasets.

3.3 RPNet

RPNet, as shown in Fig. 4, includes 3 main components: a prototypical branch, a relation branch and a relation-enhanced module. We adopt the ResMSNet network as the feature encoder to extract defect features. Then the features are passed to the two branches in parallel. The prototypical branch evaluates the samples' initial prediction by calculating the Euclidean distance of feature prototypes, and the relation branch gives the relation scores between query features and support features of each class. The relation-enhanced module integrates the calculation results obtained by two branches and provides the final classification results.

3.3.1 Prototypical branch

Fig. 4a shows the prototypical branch, which calculates the distance of query prototypes and support prototypes analogous to [18]. The branch computes a prototype $\mathbf{p}_k \in \mathbb{R}^D$, a D dimensional feature representation, of each class through the ResMSNet $f_r : \mathbb{R}^D \to \mathbb{R}^M$. In the process of feature vector embedding, in order to reduce the number of parameters, we use the global average pooling layer instead of a full connection layer. Each support prototype is the mean vector of the embedded support points belonging its class:

$$\mathbf{p}_k = \frac{1}{\mid S_k \mid} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_r(\mathbf{x}_i) \tag{3}$$

The query prototype is the feature vector after embedding:

$$\mathbf{p}_j = f_r(\mathbf{x}_j) \tag{4}$$

Then given a Euclidean distance $d : \mathbb{R}^D \times \mathbb{R}^D \to [0, +\infty)$, the branch computes the distance between $\mathbf{p}_j$ and each support prototype $\mathbf{p}_k$ as:

$$d_{k,j} = \sqrt{(\mathbf{p}_j - \mathbf{p}_k)^2} \tag{5}$$

General models based on prototypical networks generates the class distribution for a query point $\mathbf{x}_j$ based on a softmax over distances obtained above to the prototypes in the embedding space. However, for our Relation-Prototypical Network, the distances will be calculated using the output of the relation branch in the relation-enhanced module, so as to improve classification performance.

3.3.2 Relation branch

The relation branch is shown in Fig. 4b. The input feature maps of the branch are exactly the same as those input into the prototypical branch. First, each support feature map $f_r(\mathbf{x}_i)$ and query feature map $f_r(\mathbf{x}_j)$ are combined using the operator $\Delta(f_r(\mathbf{x}_i), f_r(\mathbf{x}_j))$. In this work, we assume $\Delta(,)$ to be concatenation of feature maps.

The combined feature map from the sample and query is fed into the relation module $G_r(,)$, which consists of two convolutional layers, a global average pooling layer and a sigmoid layer and eventually produces a scalar in the range of 0 to 1 representing the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, which is called the relation score. For a N-way K-shot process where $K > 1$, we element-wise sum over the embedding module outputs of all samples from each support class to form the corresponding feature map. This pooled class-level feature map is combined with the query image feature map as above so that the number of relation scores for one query is always N no matter how many samples in each support class. Finally, under the N-way K-shot setting, we generate N

relation scores for a query sample $\mathbf{x}_j$. The calculation process of relation score $r_{i,j}$ can be written as:

$$r_{i,j} = G_r(\Delta(f_r(\mathbf{x}_i), f_r(\mathbf{x}_j))), \qquad i = 1, 2, ..., N \tag{6}$$

3.3.3 Relation-enhanced module

The prototypical distances obtained by the prototypical branch can be used as the judgment basis of the class of query samples. Through the continuous training of the model, the distance between prototypes of the same category can be reduced and that between prototypes of different categories can be increased. In addition, the relation scores obtained by the relation branch can also reflect the relationship between samples. The relation score of similar samples should be close to 1, while the score of heterogeneous samples should be close to 0. We found that errors can occur in the prototypical network classification due to the distance between the heterogeneous prototypes being less than that of similar prototypes but not different enough to discriminate between the two classes; therefore we need to ensure clear discrimination between the class distances. Considering this, we decided to introduce the relation scores into the distances measurement. We take the negative logarithm of the relation score, with a value range of 0 to 1, and multiply each distance by it, which can increase the discrimination between the two classes. The classification probability p_ϕ for a query point $\mathbf{x}$ is determined using softmax:

$$p_\phi(y = k \mid \mathbf{x}) = \frac{exp(-d_{k,k} \times (-\log\, r_{k,k}))}{\sum_{k'} exp(-d_{k',k} \times (-\log\, r_{k',k}))} \tag{7}$$

The rest of RPNet is similar to ProtoNet. Learning proceeds by minimizing the negative log-probability $J(\phi)$ of the true class k via SGD. $J(\phi)$ can be written as:

$$J(\phi) = -\log\, p_\phi(y = k \mid \mathbf{x}) \tag{8}$$

Thus, a training episode simulates few-shot learning where the training episodes are formed by randomly selecting a subset of classes from the training set, then selecting a subset of examples within each class acts as the support set and a subset of remaining examples to serve as query points.

3.4 KD-RPNet

Compared with traditional few-shot learning, there is a large gap between the source domain and the target domain known as CD-FSL. This domain shift leads to poor performance of models in the target domain. Therefore, we propose KD-RPNet which utilizes unlabeled data during RPNet training and combines supervised and unsupervised learning to provide more transferable representations as illustrated in Fig. 5.

Since the source domain data input to the student network $\mathbf{x}_i^S$ is labeled, we compute the supervised cross-entropy loss as:

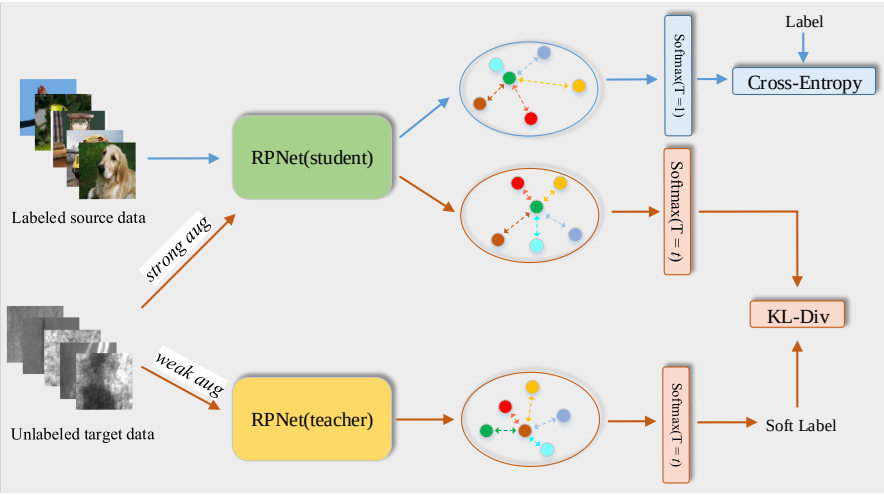


Fig. 5. KD-RPNet for cross-domain few-shot defect classification. To fairly compare the performance with other cross-domain methods, we replace ResMSNet in RPNet with ResNet10 specified in the BSCD-FSL benchmark. The student network and the teacher network are identical in structure.

$$l_{\mathrm{CE}}(y_i^S, p_i^S) = -y_i^S \log\, p_i^S \tag{9}$$

where $p_i^S = \mathrm{Softmax}(g_s(\mathbf{x}_i^S))$.

In our approach, the primary purpose of the teacher model is to provide soft labels for unlabeled data. For unlabeled samples $\mathbf{x}_j^T$, we use random-resize-crop, horizontal flip and normalization as weak augmentation (which is indicated by the superscript Tw) methods to process the images. The augmented data are denoted as $\mathbf{x}_j^{Tw}$ and then are imported into the teacher model. The prediction p_j^{Tw} produced by the teacher model is:

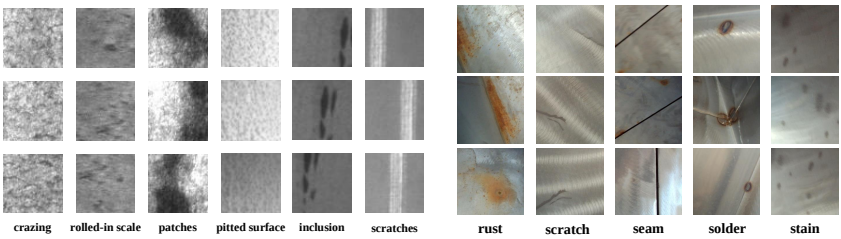
$$p_j^{Tw} = \mathrm{Softmax}(g_t(\mathbf{x}_j^{Tw})) \tag{10}$$

where p_j^{Tw} serves as the soft targets for the strongly-augmented (the superscript is Ts) unlabeled samples $\mathbf{x}_j^{Ts}$, which we obtain via the color jitter, Gaussian blur, and random gray scale transformations. The prediction p_j^{Ts} produced by the student model is:

$$p_j^{Ts} = \mathrm{Softmax}(g_t(\mathbf{x}_j^{Ts})/t) \tag{11}$$

where t is the distillation temperature, a parameter that can smooth the output probability distribution. Additionally, we use the KL divergence l_{KL} as the distillation loss, which is written as:

$$l_{\mathrm{KL}}(p_j^{Tw}, p_j^{Ts}) = \sum_{j=0}^{N^T} p_j^{Tw} \log\left(\frac{p_j^{Tw}}{p_j^{Ts}}\right) \tag{12}$$



(a) (b)

Fig. 6. (a) Examples of NEU-CLS. (b) Examples of Pipe-Defect

Finally, we update the parameters of the student RPNet f_s by minimizing the total loss function:

$$\min_{f_s} L = \frac{1}{M_S} \sum_{(\mathbf{x}_i^S, y_i^S) \in D_S} l_{\mathrm{CE}}(y_i^S, p_i^S) + \lambda \frac{1}{N_T} \sum_{\mathbf{x}_j^T \in D_T} l_{\mathrm{KL}}(p_j^{Tw}, p_j^{Ts}) \tag{13}$$

where λ is a hyper-parameter. After the training process, we evaluate the performance of the student RPNet on the target defect datasets.

4 Experiments

4.1 Datasets

In order to evaluate the effectiveness of the propsoed RPNet in few-shot defect classification, we use the benchmark dataset NEU-CLS [30] as shown in Fig. 6a. NEU-CLS is a dataset of steel surface defects collected and sorted by Northeastern University, covering 6 different types of defects with 300 images per type, including rolled-in scale, patches, crazing, pitted surface, inclusion and scratches. One of the challenges of NEU-CLS is the high similarity among the different classes. In our traditional few-shot learning experiments, we divide NEU-CLS into a training set and a testing set with a ratio of 7:3.

To effectively evaluate the performance of RPNet we benchmark with standard datasets and defect detection datasets. We the standard benchmark datasets *mini*ImageNet, *tiered*ImageNet [31] and CIFAR-FS [32]. The *mini*ImageNet dataset is the most popular few-shot learning benchmark proposed by [17] and derived from the original ILSVRC-12 [33] dataset. It contains 100 randomly sampled different categories and each with 600 images of size 84×84 pixels. The *tiered*ImageNet dataset is another few-shot learning benchmark. Like *mini*ImageNet, it is also a subset of ILSVRC-12. However, *tiered*ImageNet is a larger subset which consists of 608 classes. There are 34 categories in the dataset and the categories are divided into 20 training, 6 validation and 8 test classes, with each category contains 10 to 30 classes. The CIFAR-FS dataset is a subset of CIFAR-100 [34]. It has 100 classes divided into 64 training, 16 validation and 20 test classes and each class has 600 RGB

images with a size of 32×32 pixels. For each image, it has a fine-grained label and a coarse-grained label.

To evaluate our KD-RPNet in cross-domain few-shot defect classification tasks, we not only applied it to NEU-CLS dataset, but also independently constructed a novel few-shot ventilation pipeline defect dataset named Pipe-Defect. The examples of Pipe-Defect dataset is shown in Fig. 6b and the collection scene of dataset images is shown in Fig. 7. Pipe-Defect dataset contains 5 ventilation pipeline defect categories with 100 images each class, including rust, scratch, seam, solder and stain.

Fig. 7. The collection scene of the Pipe-Defect dataset

4.2 Experimental setting

Our RPNet and KD-RPNet models adopt an end-to-end training mode. When training, we use Adam [35] with an initial learning rate 10^{-3}, annealed by half for every 30,000 episodes and cross-entropy is used in RPNet as the loss function.

In data processing, we resized the input images from all datasets to 100×100 pixels and conducted 5-way 1-shot and 5-way 5-shot classification following the standard settings adopted by most existing few-shot learning works. As defined, the support images are 5 for 1-shot and 25 for 5-shot. Additionally, we set 15 query images for each sample class in one training episode for both 1-shot and 5-shot setting experiments. Finally, the evaluation accuracy determined within a 95% confidence interval and computed by averaging over 600 randomly sampled episodes from the testing set.

For CD-FSL, we follow the BSCD-FSL benchmark [24] to use ResNet10 as feature encoder. The model is trained on source domain *mini*ImageNet with unlabeled target data in NEU-CLS or Pipe-Defect. The distillation temperature is set to 7 specific values and λ is increased from 0 to 1 using cosine scheduling. For evaluation, we only applied the student RPNet on target

Table 1. Comparison results of few-shot defect classification accuracies(%) of ResNet12 and ResMSNet on NEU-CLS

Backbone	**ProtoNet**		**FRN**		**DeepEMD v2**	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
ResNet12	94.76 ± 0.02	96.94 ± 0.02	96.05 ± 0.11	97.98 ± 0.09	96.22 ± 0.06	98.05 ± 0.03
ResMSNet-*s*2 (Ours)	95.19 ± 0.10	97.07 ± 0.07	96.44 ± 0.07	98.16 ± 0.22	96.63 ± 0.04	98.24 ± 0.15
ResMSNet-*s*3 (Ours)	**95.68 ± 0.09**	**97.75 ± 0.12**	**96.83 ± 0.10**	**98.85 ± 0.06**	**97.15 ± 0.06**	**98.90 ± 0.19**
ResMSNet-*s*4 (Ours)	94.02 ± 0.08	96.87 ± 0.03	95.81 ± 0.04	97.63 ± 0.07	95.91 ± 0.05	97.44 ± 0.08

domain. Additionally, we used NVIDIA RTX3090 GPUs and PyTorch deep learning framework for training our networks on an Ubuntu system.

4.3 ResMSNet experimental results

For fair comparison, we use ProtoNet, FRN [36] and DeepEMD v2 [37] as the metric module respectively combined with ResNet12, a widely used backbone in few-shot classification, or ResMSNet to determine the classification results. As previously described, s is a control parameter of scale dimension. Therefore, we vary s (denote by $s2$, $s3$ and $s4$) to vary the scale dimension of ResMSNet, train the networks on NEU-CLS and compare the 5-way 1-shot and 5-way 5-shot results achieved by ResNet12 and ResMSNet in different scale dimensions. The results are shown in Table 1.

As indicated in Table 1, we can conclude that our ResMSNet outperforms ResNet12 and improves performance on NEU-CLS when its scale dimension is both 2 and 3. Overall, ResMSNet has the best performance when $s = 3$, the 5-way 1-shot classification accuracy is improved by 0.92% , 0.78%, and 0.93% respectively and the 5-way 5-shot accuracy is improved by 0.81% , 0.87%, and 0.85% respectively. ResMSNet also achieves better performance than ResNet12 when $s = 2$. However, when the scale dimension is 4, the performance of ResMSNet has decreased significantly, even worse than ResNet12. When s is large ($s = 4$), the number of sub-feature maps is large, which leads to an excessively large receptive field and hence the network's ability to extract detailed discriminative region features of defect samples is weakened. Therefore, we use ResMSNet with $s = 3$ as the multi-scale feature extraction backbone network for few-shot defect classification tasks.

Table 2. Ablation study of few-shot defect classification accuracies(%) of ResMSNet on NEU-CLS

ResMSNet	Mish	DropBlock	NEU-CLS	
			5-way 1-shot	5-way 5-shot
✓			94.57 ± 0.08	96.82 ± 0.09
✓	✓		95.12 ± 0.09	97.38 ± 0.06
✓		✓	94.83 ± 0.07	97.19 ± 0.10
✓	✓	✓	**95.68 ± 0.09**	**97.75 ± 0.12**

In addition, since we add the DropBlock to the ResMSNet block and change the activation function from ReLU to Mish, we conduct relevant ablation experiments in order to verify the effectiveness of these amendments. We still use ProtoNet as metric network and change the indicators in ResMSNet-$s3$ and the results can be seen in Table 2. The results indicate that using either the Mish activation function or DropBlock can increase the performance. When

combining both, the 1-shot accuracy is increased by 1.11% and the 5-shot performance improvement is 0.93% compared with the baseline ResMSNet.

Last but not least, we computed the number of parameters for both ResNet12 and ResMSNet backbone networks, and the results are shown in Table 3. It can be seen that our ResMSNet not only performs better, but also takes up less resources on the computation.

Table 3. Comparison results of the number of ResNet12 and ResMSNet parameters

Methods	Params(M)
ResNet12	12.42
ResMSNet	**6.52**

4.4 RPNet experimental results

4.4.1 Evaluation on defect dataset

We evaluate the performance of the proposed RPNet model by comparing with the few-shot learning baselines RelationNet [19], ProtoNet [18], MatchingNet [17] and MAML [20] using the NEU-CLS defect dataset. For a fair comparison, our model uses the well-known ResNet12 backbone. The results are summarized in Table 4. Additionally, to prove the effectiveness of the combination of ResMSNet and RPNet, another comparative experiment between RPNet with ResNet12 and RPNet with ResMSNet-$s3$ is set up. From the bottom two rows of Table 4 we can see the comparison results.

Table 4. Comparison results of few-shot defect classification accuracies(%) of RPNet and baselines on NEU-CLS

Methods	Backbone	**NEU-CLS**	
		5-way 1-shot	5-way 5-shot
MatchingNet	ResNet12	92.97 ± 0.03	94.85 ± 0.03
MAML	ResNet12	93.68 ± 0.07	95.13 ± 0.06
RelationNet	ResNet12	94.78 ± 0.04	95.62 ± 0.01
ProtoNet	ResNet12	94.76 ± 0.02	96.94 ± 0.02
RPNet (Ours)	ResNet12	**95.81 ± 0.09**	**97.68 ± 0.05**
RPNet (Ours)	ResMSNet-$s3$	**96.89 ± 0.09**	**98.73 ± 0.07**

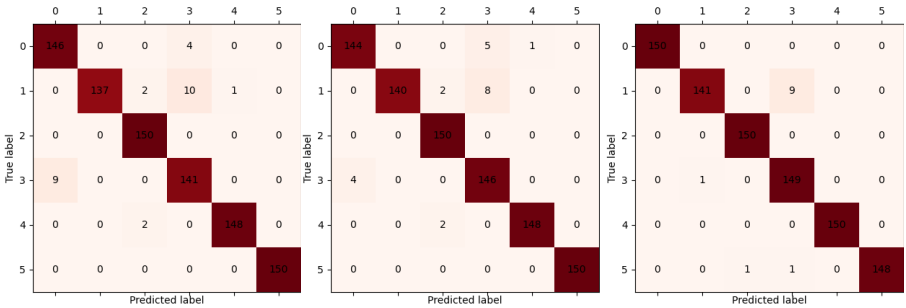


Fig. 8. Confusion matrices of classification results of (a) ProtoNet with ResNet12. (b) RPNet with ResNet12. (c) RPNet with ResMSNet-*s*3 on NEU-CLS

From the results shown in Table 4, it can be seen that the classification accuracy of our RPNet model has improved significantly which partly solves the main problem of similar color, shape and texture of the defect images in the NEU-CLS dataset compared with the baselines for few-shot classification. Particularly, when compared to ResNet12 feature encoding, our RPNet outperforms ProtoNet by 1.05% and 0.74% in terms of 5-way 1-shot and 5-way 5-shot performance, using the NEU-CLS dataset. Additionally, it also can be seen that the classification accuracy reached by RPNet with ResMSNet-*s*3 is 96.89% for 1-shot evaluation and 98.73% for 5-shot evaluation, which is 1.08% and 1.05% better than RPNet with ResNet12 respectively. Finally, to better analyze the performance of the model we proposed, we randomly draw 150 samples for each category and give the confusion matrices of the classification results of RPNet with ResNet12, RPNet with ResMSNet-s3 and ProtoNet with ResNet12 which has the best results among the baselines on NEU-CLS. Confusion matrices are shown in Fig. 8. Taking all these factors into account, we can see that either RPNet or RPNet with ResMSNet surpasses all the baselines in few-shot learning and shows the superiority in defect classification tasks.

4.4.2 Evaluation on common datasets

In addition few-shot defect classification scenarios, we proposed that our RPNet can also be applied to general few-shot classification scenarios. Hence we evaluate the performance of RPNet on three common datasets, and compare it with existing approaches. For fair comparison, we employ RPNet with the same backbones as other methods. We divide the methods into two groups according to ResNet12 and ConvNet-64. What's more, in our opinion, the ability of ResMSNet to extract tiny and discriminative features is also effective in improving the performance of RPNet in common datasets. In order to verify the robustness of ResMSNet in different scenarios, we also evaluate RPNet with

Table 5. Comparison results of few-shot classification accuracies(%) on *mini*ImageNet

Methods	Backbone	***mini*ImageNet**	
		5-way 1-shot	5-way 5-shot
BOIL [38]	ConvNet-64	49.61 ± 0.16	66.45 ± 0.37
OVE PG G P+Cosine [39]	ConvNet-64	48.00 ± 0.24	67.14 ± 0.23
IMP [40]	ConvNet-64	49.60 ± 0.80	68.10 ± 0.80
Arcmax [41]	ConvNet-64	51.90 ± 0.79	69.07 ± 0.59
RPNet (Ours)	ConvNet-64	**51.93 ± 0.64**	**70.63 ± 0.45**
MetaGAN [42]	ResNet12	52.71 ± 0.64	68.63 ± 0.67
AdaResNet [43]	ResNet12	56.88 ± 0.62	71.94 ± 0.57
PPA [44]	ResNet12	59.60 ± 0.41	73.74 ± 0.19
RPNet (Ours)	ResNet12	**59.85 ± 0.65**	**75.20 ± 0.24**
RPNet (Ours)	ResMSNet-*s*3	**60.88 ± 0.39**	**76.87 ± 0.43**

ResMSNet-s3 on each dataset. The experimental results on *mini*ImageNet, *tiered*ImageNet and CIFAR-FS are summarized in Table 5, 6 and 7.

Table 6. Comparison results of few-shot classification accuracies(%) on *tiered*ImageNet

Methods	Backbone	***tiered*ImageNet**	
		5-way 1-shot	5-way 5-shot
Ravichandran et al. [45]	ConvNet-64	48.19 ± 0.43	65.50 ± 0.39
BOIL	ConvNet-64	49.35 ± 0.26	69.37 ± 0.12
ProtoNet	ConvNet-64	53.34 ± 0.89	72.69 ± 0.74
RPNet (Ours)	ConvNet-64	**54.67 ± 0.90**	**73.93 ± 0.61**
TPN [15]	ResNet12	59.91 ± 0.94	73.30 ± 0.75
TapNet [46]	ResNet12	63.08 ± 0.15	80.26 ± 0.12
Meta-Transfer [47]	ResNet12	65.62 ± 1.80	80.61 ± 0.90
RPNet (Ours)	ResNet12	**65.89 ± 0.34**	**80.83 ± 0.76**
RPNet (Ours)	ResMSNet-*s*3	**66.73 ± 0.23**	**83.11 ± 0.42**

Table 7. Comparison results of few-shot classification accuracies(%) on CIFAR-FS

Methods	Backbone	CIFAR-FS	
		5-way 1-shot	5-way 5-shot
R2D2 [32]	ConvNet-64	65.3 ± 0.2	79.4 ± 0.1
SIB [48]	ConvNet-64	68.7 ± 0.6	77.1 ± 0.4
Wang et al. [49]	ConvNet-64	64.2± 0.3	78.4± 0.3
ConstellationNet [50]	ConvNet-64	69.3 ± 0.3	82.7 ± 0.2
RPNet (Ours)	ConvNet-64	**71.4 ± 0.3**	**85.9 ± 0.4**
TEAM [51]	ResNet12	70.43	81.25
MetaOptNet [52]	ResNet12	72.0 ± 0.7	84.2 ± 0.5
ICI [53]	ResNet12	73.97	84.13
NCA nearest centroid [54]	ResNet12	72.49± 0.12	85.15± 0.10
Curvature Generation [55]	ResNet12	73.0± 0.7	85.8± 0.5
RPNet (Ours)	ResNet12	**74.0 ± 0.5**	**86.9 ± 0.8**
RPNet (Ours)	ResMSNet-*s*3	**75.6 ± 0.7**	**87.6 ± 0.3**

As the experimental results show in Table 5, 6, and 7, our RPNet shows strong competitiveness on common datasets when compared with existing methods. In Table 5, The performance shows superiority, demonstrating up to 1.56% improvement over other methods for 5-way 5-shot evaluation and a 0.25% improvement for 5-way 1-shot evaluation. Additionally, it can be seen from Table 6 that the classification accuracies of RPNet under the two settings exceed those of existing methods by a maximum of 1.33% and 1.24%. The CIFAR-FS experimental results shown in Table 7 indicate that our accuracies exceed other state-of-the-art methods by 2.1% and 3.2%. Additionally, the performance of RPNet with ResMSNet-s3 surpass RPNet with ResNet12 on no matter which dataset. Given these results, our RPNet method proves its effectiveness for general few-shot learning classification tasks and ResMSNet is a robust backbone.

4.5 KD-RPNet experimental results

4.5.1 Find the best distillation temperature

Before comparing with state-of-the-art methods, we need to obtain a KD-RPNet model with the best cross-domain effect. In the network structure of KD-RPNet there is a parameter, distillation temperature t, which can affect the cross-domain performance. We change t and conduct extensive experiments

using NEU-CLS and Pipe-Defect to determine the value of t that provides the best cross-domain defect classification accuracy. We train KD-RPNet with a ResNet10 backbone using *mini*ImageNet and evaluate the student RPNet on target defect datasets. In addition, we set a controlled trial with a simple RPNet respectively. Experimental results are shown in Fig. 9.

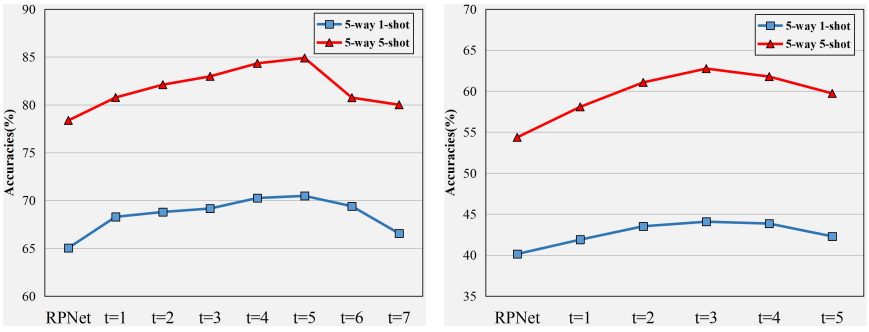


Fig. 9. (a) Results of RPNet (Column 1) and KD-RPNet in different t (Column 2 to 8) on NEU-CLS. (b) Results of RPNet (Column 1) and KD-RPNet in different t (Column 2 to 6) on Pipe-Defect

Fig. 9a shows that $t = 5$ is the optimal distillation temperature for NEU-CLS. The cross-domain classification performance of simple RPNet is not as good as KD-RPNet when $t = 1$ and when t is from 1 to 5, the performance gradually increases. However, when t is greater than 5, we can find the accuracy has decreased. The maximum t shown in the figure is 7, but in fact we increased t to 10 during experiments, and the performance keeps declining. So we believe that the accuracy is monotonically decreasing if t exceeds the maximum value. Experiments on Pipe-Defect are similar and we conclude that our KD-RPNet can achieve the highest cross-domain classification accuracy when $t = 3$ as Fig. 9b shown. In the subsequent comparison experiments, we use KD-RPNet model with the best performance by default to participate in the comparison.

4.5.2 Comparison to state-of-the-art

Table 8 shows the comparison results of our proposed methods with other state-of-the-art methods. As indicated in BSCD-FSL, all models use ResNet10 as the backbone network and are trained on *mini*ImageNet (source domain). Apart from this our KD-RPNet is fed with the same amount of unlabeled target data as the source data during training.

Our KD-RPNet has not only made improvements relative to our RPNet, but also outperformed almost all state-of-the-art approaches at all settings. Using NEU-CLS, we achieve a maximum of 4.66% improvement for 1-shot. For the 5-shot evaluation we achieve a more significant improvement of 5.5%.

Table 8. Cross-domain few-shot defect classification accuracies(%) and average inference time(in seconds) for each 5-shot task on NEU-CLS and Pipe-Defect and comparison of the number of parameters

Methods	Backbone	NEU-CLS		Pipe-Defect		Inference Time	Params(M)
		5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot		
FT-MatchingNet [56]	ResNet10	67.70±0.65	80.85±0.51	39.89±0.35	55.69±0.34	-	10.43
LRP-RelationNet [57]		65.82±0.55	79.38±0.40	-	-	-	**12.17**
TPN+ATA [58]		69.38±0.48	79.75±0.35	40.96±0.34	55.24±0.31	0.147	7.46
GNN+ATA [58]		67.63±0.46	80.18±0.36	42.54±0.33	60.07±0.29	0.205	5.49
Meta-FDMixup [59]		69.58±0.66	84.31±0.43	-	-	0.134	5.45
GNN+AFA [60]		69.91±0.44	83.48±0.34	43.06±0.42	62.23±0.39	0.073	5.50
TPN+AFA [60]		70.33±0.47	81.99±0.41	43.74±0.39	61.75±0.33	0.042	7.46
ME-D2N [61]		69.52±0.68	**86.07±0.32**	42.99±0.48	63.74±0.28	**0.017**	10.69
RPNet (Ours)	ResNet10	65.03±0.17	78.37±0.85	40.14±0.26	54.37±0.22	-	-
KD-RPNet (Ours)		70.48±0.15	84.90±0.33	44.08±0.35	62.76±0.46	0.027	10.48
KD-RPNet (Ours)	ResMSNet-*s*3 (Ours)	**71.60±0.44**	85.75±0.26	**45.17±0.44**	**63.8±0.24**	-	-

Among the approaches evaluated, the accuracy of ME-D2N is closest to KD-RPNet, but we still outperform it by 0.96% for 1-shot. Using Pipe-Defect, we obtain similar results as when using NEU-CLS. Compared with ATA and FT methods, two methods introduced in 2021, we achieve 1.54% average improvement for 1-shot and 2.69% average improvement for 5-shot. As for methods proposed in 2022, our KD-RPNet is better than AFA and comparable to ME-D2N. We also set up experiments of KD-RPNet with ResMSNet-$s3$ and it outperforms KD-RPNet with ResNet10. In order to better reflect the performance improvement of KD-RPNet, we randomly draw 150 samples on NEU-CLS for each class and give the confusion matrices of RPNet and KD-RPNet experimental results as shown in Fig. 10. These results give us reason to believe that our approaches can apply to cross-domain few-shot defect classification tasks and achieve state-of-the-art performance.

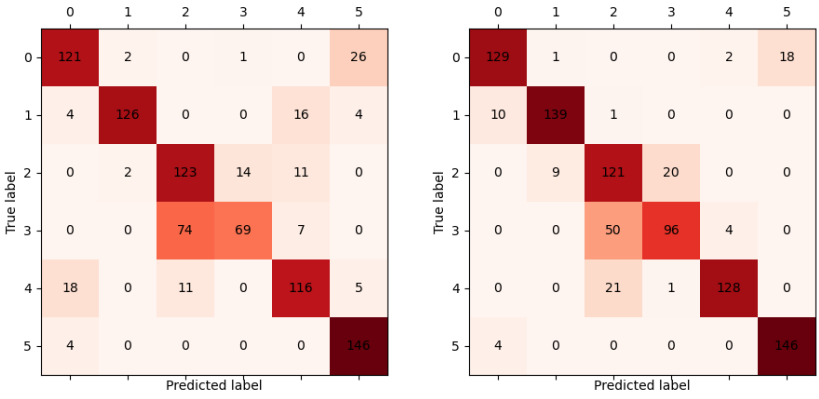
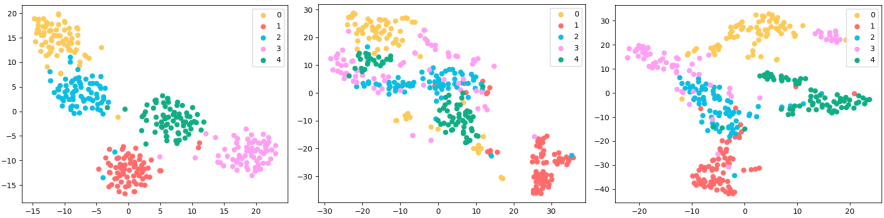


Fig. 10. Confusion matrices of classification results of (a) RPNet. (b) KD-RPNet on NEU-CLS

We further conduct inference time experiments to investigate the computation efficiency of KD-RPNet and other state-of-the-art methods. We compute the average inference time required for each 5-shot task on NEU-CLS and Pipe-Defect. The results can also be seen in Table 8. It shows that our model not only performs better than other recent models but also costs less time except for ME-D2N.

4.5.3 Analysis and visualization

In order to explore the reasons why our method can improve the classification performance of the target domain, we use the model pretrained on *mini*ImageNet to extract features of the same domain and NEU-CLS separately. We then use KD-RPNet, whose source domain is *mini*ImageNet, and the unlabeled target domain NEU-CLS to extract defect features. We plot the



(a) (b) (c)

Fig. 11. T-SNE visualization of RPNet training on *mini*ImageNet and (a) evaluation on *mini*ImageNet; (b) evaluation on NEU-CLS; (c) KD-RPNet evaluation with *mini*ImageNet as source domain and NEU-CLS as unlabeled target domain

learned features with t-SNE [62]. The number of clusters is set to be 5 random classes of the target domain. Fig. 11 shows the t-SNE visualization of the three cases. It can be seen from Fig 11a that in-domain classification has a good clustering effect. However, under CD-FSL the clustering performance is greatly reduced as shown in Fig. 11b. In Fig. 11c we see the effect of including unlabeled target data in the model training process where our model learns better clusters than before. By comparing the embeddings and analysing, we see that even though we do not use any labels for target data during training, our model can learn more specific representations from target domain, which is good for improving cross-domain few-shot classification performance.

Although our KD-RPNet achieves state-of-the-art level in cross-domain few-shot defect classification tasks, there are still some shortcomings and challenges. As the last column of Table 8 shown is the number of KD-RPNet and other state-of-the-art methods parameters. It can be seen that the number of our KD-RPNet parameters is not the most, but still quite a few. Since our KD-RPNet is a knowledge distillation-based method that aims to learn more target specific representations while learning common embedding. It contains a teacher network and a student network, this leads to a relatively high number of parameters. At present, using target domain embeddings to guide network training has been considered to be a very effective method, and the biggest challenge we face is how to build a more effective and lightweight bridge between target domain features and source domain features so as to improve the performance of the network while avoiding a large increase in the amount of network parameters.

5 Conclusion

In this paper, in order to tackle the current problems in defect classification, we develop a novel backbone namely ResMSNet, which aims to focus on tiny discriminative defect features. We then propose a Relation-Prototypical Network (RPNet) which improves classification performance by using relation

scores to adjust prototype distances. Finally, for CD-FSL, a more realistic scenario, we improve RPNet with the idea of knowledge distillation and introduce KD-RPNet. By utilizing unlabeled target defect data we demonstrate our model can learn more specific representations. Extensive ablation and comparative experiments show the effectiveness of ResMSNet and our RPNet in outperforming baselines on NEU-CLS. In addition, we construct a novel dataset, Pipe-Defect and use this to further evaluate the approaches. Experiments on the novel dataset and NEU-CLS prove that KD-RPNet provides a state-of-the-art approach in cross-domain few-shot defect classification.

Conflict of interest statement

We declare that we have no financial and personal relationships that can directly or indirectly influence the work submitted.

Data available

The dataset generated during the current study is not public because it contains proprietary information obtained by the authors through a license. Information on how to obtain it is available from the corresponding author upon reasonable request.

References

[1] Gao, S.-H., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., Torr, P.: Res2net: A new multi-scale backbone architecture. IEEE transactions on pattern analysis and machine intelligence **43**(2), 652–662 (2019)

[2] Neogi, N., Mohanta, D.K., Dutta, P.K.: Review of vision-based steel surface inspection systems. EURASIP Journal on Image and Video Processing **2014**(1), 1–19 (2014)

[3] Kaftandjian, V., Zhu, Y.M., Dupuis, O., Babot, D.: The combined use of the evidence theory and fuzzy logic for improving multimodal nondestructive testing systems. IEEE Transactions on Instrumentation and Measurement **54**(5), 1968–1977 (2005)

[4] Çelik, H., Dülger, L., Topalbekiroğlu, M.: Development of a machine vision system: real-time fabric defect detection and classification with neural networks. The Journal of The Textile Institute **105**(6), 575–585 (2014)

[5] Yi, L., Li, G., Jiang, M.: An end-to-end steel strip surface defects recognition system based on convolutional neural networks. steel research international **88**(2), 1600068 (2017)

[6] Zhang, J., Su, H., Zou, W., Gong, X., Zhang, Z., Shen, F.: Cadn: a weakly supervised learning-based category-aware object detection network for surface defect detection. Pattern Recognition **109**, 107571 (2021)

[7] Tabernik, D., Šela, S., Skvarč, J., Skočaj, D.: Segmentation-based deep-learning approach for surface-defect detection. Journal of Intelligent Manufacturing **31**(3), 759–776 (2020)

[8] Wei, B., Hao, K., Gao, L., Tang, X.-S.: Bioinspired visual-integrated model for multilabel classification of textile defect images. IEEE Transactions on Cognitive and Developmental Systems **13**(3), 503–513 (2020)

[9] Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence **28**(4), 594–611 (2006)

[10] Lake, B., Salakhutdinov, R., Gross, J., Tenenbaum, J.: One shot learning of simple visual concepts. In: Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 33 (2011)

[11] Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science **350**(6266), 1332–1338 (2015)

[12] Ge, W., Yu, Y.: Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1086–1095 (2017)

[13] Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., Feris, R.: Spottune: transfer learning through adaptive fine-tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4805–4814 (2019)

[14] Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.-S., Schiele, B.: Learning to self-train for semi-supervised few-shot classification. Advances in Neural Information Processing Systems **32** (2019)

[15] Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S.J., Yang, Y.: Learning to propagate labels: Transductive propagation network for few-shot learning. arXiv preprint arXiv:1805.10002 (2018)

[16] Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K.: Semi-supervised domain adaptation via minimax entropy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8050–8058 (2019)

[17] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in neural information processing systems **29** (2016)

[18] Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)

[19] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1199–1208 (2018)

[20] Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning, pp. 1126–1135 (2017). PMLR

[21] Bao, Y., Song, K., Liu, J., Wang, Y., Yan, Y., Yu, H., Li, X.: Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. IEEE Transactions on Instrumentation and Measurement **70**, 1–11 (2021)

[22] Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C.F., Huang, J.-B.: A closer look at few-shot classification. arXiv preprint arXiv:1904.04232 (2019)

[23] Tseng, H.-Y., Lee, H.-Y., Huang, J.-B., Yang, M.-H.: Cross-domain

few-shot classification via learned feature-wise transformation. In: International Conference on Learning Representations (2020)

[24] Guo, Y., Codella, N.C., Karlinsky, L., Codella, J.V., Smith, J.R., Saenko, K., Rosing, T., Feris, R.: A broader study of cross-domain few-shot learning. In: European Conference on Computer Vision, pp. 124–141 (2020). Springer

[25] Sa, L., Yu, C., Chen, Z., Zhao, X., Yang, Y.: Attention and adaptive bilinear matching network for cross-domain few-shot defect classification of industrial parts. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2021). IEEE

[26] Xiao, W., Song, K., Liu, J., Yan, Y.: Graph embedding and optimal transport for few-shot classification of metal surface defect. IEEE Transactions on Instrumentation and Measurement **71**, 1–10 (2022)

[27] Islam, A., Chen, C.-F.R., Panda, R., Karlinsky, L., Feris, R., Radke, R.J.: Dynamic distillation network for cross-domain few-shot recognition with unlabeled data. Advances in Neural Information Processing Systems **34**, 3584–3595 (2021)

[28] Misra, D.: Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681 (2019)

[29] Ghiasi, G., Lin, T.-Y., Le, Q.V.: Dropblock: A regularization method for convolutional networks. Advances in neural information processing systems **31** (2018)

[30] Song, K., Yan, Y.: A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Applied Surface Science **285**, 858–864 (2013)

[31] Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. arXiv preprint arXiv:1803.00676 (2018)

[32] Bertinetto, L., Henriques, J.F., Torr, P.H., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. arXiv preprint arXiv:1805.08136 (2018)

[33] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.*: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)

[34] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features

from tiny images (2009)

[35] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[36] Wertheimer, D., Tang, L., Hariharan, B.: Few-shot classification with feature map reconstruction networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8012–8021 (2021)

[37] Zhang, C., Cai, Y., Lin, G., Shen, C.: Deepemd: Differentiable earth mover's distance for few-shot learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)

[38] Oh, J., Yoo, H., Kim, C., Yun, S.-Y.: Boil: Towards representation change for few-shot learning. In: International Conference on Learning Representations (2020)

[39] Snell, J., Zemel, R.: Bayesian few-shot classification with one-vs-each pólya-gamma augmented gaussian processes. In: International Conference on Learning Representations (2020)

[40] Allen, K., Shelhamer, E., Shin, H., Tenenbaum, J.: Infinite mixture prototypes for few-shot learning. In: International Conference on Machine Learning, pp. 232–241 (2019). PMLR

[41] Afrasiyabi, A., Lalonde, J.-F., Gagné, C.: Associative alignment for few-shot image classification. In: European Conference on Computer Vision, pp. 18–35 (2020). Springer

[42] Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., Song, Y.: Metagan: An adversarial approach to few-shot learning. Advances in neural information processing systems **31** (2018)

[43] Munkhdalai, T., Yuan, X., Mehri, S., Trischler, A.: Rapid adaptation with conditionally shifted neurons. In: International Conference on Machine Learning, pp. 3664–3673 (2018). PMLR

[44] Qiao, S., Liu, C., Shen, W., Yuille, A.L.: Few-shot image recognition by predicting parameters from activations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7229–7238 (2018)

[45] Ravichandran, A., Bhotika, R., Soatto, S.: Few-shot learning with embedded class models and shot-free meta training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 331–339 (2019)

[46] Yoon, S.W., Seo, J., Moon, J.: Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In: International Conference on Machine Learning, pp. 7115–7123 (2019). PMLR

[47] Sun, Q., Liu, Y., Chua, T.-S., Schiele, B.: Meta-transfer learning for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 403–412 (2019)

[48] Hu, S.X., Moreno, P.G., Xiao, Y., Shen, X., Obozinski, G., Lawrence, N.D., Damianou, A.: Empirical bayes transductive meta-learning with synthetic gradients. arXiv preprint arXiv:2004.12696 (2020)

[49] Wang, Z., Miao, Z., Zhen, X., Qiu, Q.: Learning to learn dense gaussian processes for few-shot learning. Advances in Neural Information Processing Systems **34**, 13230–13241 (2021)

[50] Xu, W., Wang, H., Tu, Z., *et al.*: Attentional constellation nets for few-shot learning. In: International Conference on Learning Representations (2020)

[51] Qiao, L., Shi, Y., Li, J., Wang, Y., Huang, T., Tian, Y.: Transductive episodic-wise adaptive metric for few-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3603–3612 (2019)

[52] Lee, K., Maji, S., Ravichandran, A., Soatto, S.: Meta-learning with differentiable convex optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10657–10665 (2019)

[53] Wang, Y., Xu, C., Liu, C., Zhang, L., Fu, Y.: Instance credibility inference for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12836–12845 (2020)

[54] Laenen, S., Bertinetto, L.: On episodes, prototypical networks, and few-shot learning. Advances in Neural Information Processing Systems **34**, 24581–24592 (2021)

[55] Gao, Z., Wu, Y., Jia, Y., Harandi, M.: Curvature generation in curved spaces for few-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8691–8700 (2021)

[56] Tseng, H.-Y., Lee, H.-Y., Huang, J.-B., Yang, M.-H.: Cross-domain few-shot classification via learned feature-wise transformation. In: International Conference on Learning Representations (2020)

[57] Sun, J., Lapuschkin, S., Samek, W., Zhao, Y., Cheung, N.-M., Binder, A.:

Explanation-guided training for cross-domain few-shot classification. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 7609–7616 (2021). IEEE

[58] Wang, H., Deng, Z.-H.: Cross-domain few-shot classification via adversarial task augmentation. arXiv preprint arXiv:2104.14385 (2021)

[59] Fu, Y., Fu, Y., Jiang, Y.-G.: Meta-fdmixup: Cross-domain few-shot learning guided by labeled target data. arXiv preprint arXiv:2107.11978 (2021)

[60] Hu, Y., Ma, A.J.: Adversarial feature augmentation for cross-domain few-shot classification. arXiv preprint arXiv:2208.11021 (2022)

[61] Fu, Y., Xie, Y., Fu, Y., Chen, J., Jiang, Y.-G.: Me-d2n: Multi-expert domain decompositional network for cross-domain few-shot learning. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 6609–6617 (2022)

[62] Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)