Research Article

Loss Architecture Search for Few-Shot Object Recognition

Jun Yue ^{1,2}, Zelang Miao ³, Yueguang He,¹ and Nianchun Du⁴

¹Department of Geomatics Engineering, School of Traffic & Transportation Engineering,
Changsha University of Science & Technology, Changsha, China

²Institute of Remote Sensing and Geographic Information System, Peking University, Beijing, China

³School of Geosciences and Info-Physics, Central South University, Changsha, China

⁴China Nonferrous Metal Changsha Survey and Design Institute, Changsha, China

Correspondence should be addressed to Zelang Miao; zelang.miao@csu.edu.cn

Received 12 January 2020; Revised 10 July 2020; Accepted 13 July 2020; Published 24 August 2020

Academic Editor: Alex Alexandridis

Copyright © 2020 Jun Yue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Few-shot object recognition, which exploits a set of well-labeled data to build a classifier for new classes that have only several samples per class, has received extensive attention from the machine learning community. In this paper, we investigate the problem of designing an optimal loss function for few-shot object recognition and propose a novel few-shot object recognition system that includes the following three steps: (1) generate a loss function architecture using a recurrent neural network (generator); (2) train a base embedding network with the generated loss function on a training set; (3) fine-tune the base embedding network using the few-shot instances from a validation set to obtain the accuracy and use it as a reward signal to update the generator. This procedure is repeated and implemented in the reinforcement learning framework for finding the best loss architecture such that the embedding network yields the highest validation accuracy. Our key insight is to create a search space of the loss function architectures and evaluate the quality of a particular loss function on the dataset of interest. We conduct experiments on three popular datasets for few-shot learning. The results show that the proposed approach achieves better performance than state-of-the-art methods.

1. Introduction

The object recognition problem under conventional supervised learning has been thoroughly studied with many successful models available, especially deep convolutional neural networks (DCNNs), including VGG [1], GoogLeNet [2], ResNet [3], and NASNet [4]. DCNNs can achieve impressive performance on large datasets like ImageNet [5] and Open Images [6]. However, in order to train a DCNN model, large amounts of labeled training data and training time are typically needed. Some works use pretrained DCNNs as off-the-shelf deep feature extractors [7]. While this procedure can reduce training time, it still requires lots of labeled images to fine-tune the models. This is in contrast with how a human vision recognition system works: a person does not need to see thousands of instances of an object, but only a small number of them, to remember and generalize for the recognition of the object later [8].

Motivated by the high capability of human vision recognition system, one-/few-shot learning has attracted considerable attention from the computer vision community, including those on one-shot or few-shot learning for object recognition [9–14], object detection [15], image segmentation [16], image captioning [17], face identification [18], and person reidentification [19]. Different from the standard supervised learning, few-shot learning aims to solve the problem where only a few labeled samples are available for training, while its extreme scenario, called one-shot learning, handles the more challenging situation where only one instance per class is available. As training a classifier using only one or a few instances per class is difficult to achieve a desired result, studies on one-/few-shot object recognition have focused on training it with a set of well-labeled data and then generalizing it to new classes [10, 20].

For one-/few-shot object recognition, due to the scarcity of the labeled examples from each new class, a natural metric learning scheme is adopted in [21], which aims to learn

object representation and categorize instances based on the absolute similarities between pairs of instances. This absolute similarity-based method, however, does not pay enough attention to the interclass and intraclass correlation, which may not result in a satisfactory accuracy.

Instead of absolute similarity, the scheme of relative similarity-based classification can have higher capability in decreasing the intraclass variations and increasing the interclass variations [22]. While it is difficult to separate or group objects only based on an absolute similarity measure, relative similarities can conveniently deal with these variations by requiring the intraclass similarity to be larger than the interclass similarity. Figure 1 shows an example of the advantage of relative similarity. There have been many metric learning methods based on relative similarity [23–26]. However, different metric learning losses can lead to different performances and there has been no rule of thumb to design the best loss for a dataset.

Hence, in this paper, we develop a novel object representation-based metric learning method that induces suitable relative similarities with a particular loss generated by loss architecture search (LAS), inspired by the neural architecture search (NAS) [4, 27]. LAS, as shown in Figure 2, is a gradient-based method for finding good loss architectures. We use a recurrent neural network as the loss function architecture generator to generate a suitable loss for training the embedding network. The generator is updated with a reward signal, which measures how good the metric learning loss is based on the relative similarity. To obtain the reward signal, the following two steps are carried out. First, we train the embedding network with the generated loss L using the training set. Second, we use the few-shot instances from the validation set to fine-tune the embedding network to obtain the accuracy R . With this accuracy as the reward signal, we can compute the policy gradient (i.e., the gradient of P) by proximal policy optimization (PPO) [28] to update the loss generator. As a result, in the next iteration, the generator gives higher probabilities to those losses that drive the embedding network to obtain higher accuracies. In other words, the generator learns to improve its search over time. After obtaining the final loss function L^* with the highest validation accuracy, the few-shot object recognition system is completed by updating the embedding network with the training set and the testing set (see the right part of Figure 2).

The proposed approach trains the object embedding network on the well-labeled training set under a loss function generated by LAS, such that the instances from new classes can be categorized based on their similarities with the few-shot instances in the learned embedding space. Our main contributions are threefold:

- (1) We create a search space for automatically determining the hyperparameters of metric learning losses, including their magnitudes, margins, and angle
- (2) We present a few-shot recognition system where a proximal policy optimization algorithm is used to

find the best loss architecture sampled from the search space such that the embedding network yields the highest validation accuracy

- (3) Our few-shot object recognition system with LAS by reinforcement learning obtains the best results compared with recent state-of-the-art methods

2. Related Work

2.1. Few-Shot Learning. Recent works for few-shot learning have adopted different strategies in different domains [8, 9, 11, 18, 21, 29–33], among which the most relevant ones are the metric learning-based methods [8, 21]. The Siamese network developed in [21] aims to learn absolute similarity between pairs of objects. It enlarges or lessens the similarity between a pair of samples according to whether they are from the same class. The authors of [8] propose a matching network which can also be considered as a metric learning-based method. This approach minimizes the cosine distance based-Siamese loss between the embeddings of instances with a bidirectional Long Short-Term Memory (bi-LSTM). Another line of research uses metalearning for few-shot learning, which trains a *metalearner* from many relevant tasks. In [29], model agnostic metalearning is proposed to train a model that can quickly adapt to a new task with limited training data. The authors of [9] propose an LSTM-based *metalearner* model to train another learner in the few-shot regime. Another metalearning-based method is proposed in [30] which learns to fast parameterize an underlying neural network by employing metainformation. The memory-augmented neural network is adopted in [31] for few-shot learning. It enables each instance to be encoded and retrieved efficiently by a memory module. This module consists of key-value pairs where keys and values are the embeddings and labels of examples, respectively. Another work from [32] also uses a memory module for training, with a bi-LSTM to predict the parameters of the embedding model. The authors of [33] propose a task agnostic approach to improve the generalizability of few-shot metalearning. In addition to the research mentioned above, a few other works generate synthetic data for learning [11, 18]. Different from these previous methods, we design our few-shot object recognition system in the reinforcement learning framework, which automatically determines the optimal loss function for training the embedding network.

2.2. Metric Learning. Metric learning aims to learn a similarity function from examples, which measures the similarity between object pairs. Recently, quite a few methods have been proposed for metric learning. Some works [34–36] adopt pairwise constraints to train the models. As discussed in Section 1, these models’ capacity is limited due to the use of absolute similarity. Most recent works [23, 26, 37] use DCNNs as embedding functions and use triplet-based constraints instead of pairwise constraints to capture similarities between objects. Their results show that the combination of a deep embedding model with a triplet-based constraint is effective in learning similarity functions.

FIGURE 1: An example that shows the advantage of relative similarity. Left: nine objects from three different classes. The lemons and oranges are similar in appearance. Middle: a pairwise similarity-based model may be confused by a pair of objects from the lemon and orange classes. Right: if a model learns relative similarity between objects, it may not be confused since the anchor object (in the blue box) is relatively similar to the lemon compared to the orange.

FIGURE 2: Overview of the proposed loss architecture search (LAS) for few-shot object recognition.

Nevertheless, the problems solved in these works still fall into the conventional supervised learning setting where a set of well-labeled data are needed. In our work, instead of manually designing a metric learning loss and training with a large amount of labeled data, we use LAS for metric learning, which can automatically determine the hyperparameters of relative similarity-based losses such as margin, magnitude, and angle.

2.3. AutoML. The success of deep learning in computer vision tasks is largely due to its automation of the feature extraction process; hierarchical feature extractors are learned in an end-to-end fashion from data rather than being manually designed. To obtain better performance and handle more difficult problems, researchers have manually developed more and more complex networks. To automate the design of network architectures, neural architecture search [4, 27] is thus a logical next step in machine learning. NAS can be seen as a subfield of AutoML. The work form [38] proposed a method to automatically determine the parameters of softmax-based loss architecture by using AutoML. In our work, we propose LAS for finding the best architecture of the metric learning-based loss function for few-shot learning, which can be seen as another subfield of AutoML. We believe that combining

NAS and LAS will make AutoML go further and this is the future work.

3. Loss Architecture Search

We formulate the problem of finding the best loss architecture as a discrete search problem. In our search space, a loss architecture consists of five subloss architectures, each of which has two hyperparameters: (1) the magnitude of the operation and (2) the margin or angle of the subloss.

3.1. Losses. We choose five promising losses to construct the search space: Hard Mining Triplet Loss (HMTL) [23], Margin Sample Mining Loss (MSML) [24], Angular Loss (AL) [25], Triplet Center Loss (TCL) [26], and Quadruplet Center Loss (QCL). The first four are from the metric learning community, which achieve state-of-the-art results in few-shot learning, face recognition, vehicle Re-ID, person Re-ID, and so forth. The last one, QCL, is proposed in this paper and is a complement to the first four losses.

3.1.1. Hard Mining Triplet Loss. The triplet loss is based on a relative distance comparison among triplet instances. Specifically, for any triplet $(x_a^{(i)}, x_p^{(i)}, x_n^{(i)})$ where $x_a^{(i)}$ and $x_p^{(i)}$ are two distinct samples of the same class and $x_n^{(i)}$ comes from

another class as a negative example, their embedding vectors are generated by an embedding function f_ϕ such that $h_a^{(i)} = f_\phi(x_a^{(i)})$, $h_p^{(i)} = f_\phi(x_p^{(i)})$, and $h_n^{(i)} = f_\phi(x_n^{(i)})$. The relative distance constraint of each triplet is defined as

$$D(h_a^{(i)}, h_p^{(i)}) + m \leq D(h_a^{(i)}, h_n^{(i)}), \quad (1)$$

where m is a predefined constant parameter representing the minimum margin between the matched and mismatched pairs, $D(x, y)$ is a metric function that measures distance between vectors x and y , and i denotes the i -th triplet. In order to ensure fast convergence, it is crucial to select triplets that violate the triplet constraint in equation (1). This means that, given $x_a^{(i)}$, we want to select an $x_p^{(i)}$ (hard positive sample) such that $x_p^{(i)} = \operatorname{argmax}_{x_p} D(f_\phi(x_a^{(i)}), f_\phi(x_p))$ and to select $x_n^{(i)}$ (hard negative sample) such that $x_n^{(i)} = \operatorname{argmin}_{x_n} D(f_\phi(x_a^{(i)}), f_\phi(x_n))$. For each sample (as an anchor) from a batch of size M , we select a hard-positive sample and a hard-negative sample (also from the batch) to construct a triplet. Thus, the HTML function can be defined as

$$L_{\text{HTML}} = \frac{1}{M} \sum_{i=1}^M \max(0, m + D(h_a^{(i)}, h_p^{(i)}) - D(h_a^{(i)}, h_n^{(i)})). \quad (2)$$

3.1.2. Margin Sample Mining Loss. The quadruplet loss (QL) [22] extends the triplet loss by adding a different negative pair¹. A quadruplet contains four different instances $\{x_a^{(i)}, x_p^{(i)}, x_{n_1}^{(i)}, x_{n_2}^{(i)}\}$, where $x_a^{(i)}$ and $x_p^{(i)}$ are samples of the same class, while $x_{n_1}^{(i)}$ and $x_{n_2}^{(i)}$ are samples of two other classes. The quadruplet loss is formulated as

$$L_{\text{QL}} = \frac{1}{N^{\text{QL}}} \sum_{i=1}^{N^{\text{QL}}} \max(0, m_1 + D(h_a^{(i)}, h_p^{(i)}) - D(h_a^{(i)}, h_{n_1}^{(i)})) + \frac{1}{N^{\text{QL}}} \sum_{i=1}^{N^{\text{QL}}} \max(0, m_2 + D(h_a^{(i)}, h_p^{(i)}) - D(h_{n_1}^{(i)}, h_{n_2}^{(i)})), \quad (3)$$

where m_1 and m_2 are two margins for the two terms, respectively, and N^{QL} stands for the number of quadruplets. The first term is the same as the triplet loss. The second term tries to enforce intraclass distances to be smaller than interclass distances [22]. Based on QL, a strategy named margin sample mining is proposed [24]. It picks the most dissimilar positive pair (x_a, x_p) and the most similar negative pair (x_{n_1}, x_{n_2}) in the whole batch and formulates its loss as

$$L_{\text{MSML}} = \max(0, m + D(f_\phi(x_a), f_\phi(x_p)) - D(f_\phi(x_{n_1}), f_\phi(x_{n_2}))). \quad (4)$$

As shown in equation (4), the constraints of MSML are extremely sparse. Only two pairs in a batch are used to compute the gradient of the embedding network in the

training phase, and it seems that it wastes a lot of training data. In fact, the two chosen pairs are determined by all the data in one batch. During training, more and more pairs will be selected [24].

3.1.3. Angular Loss. Let a triplet $(x_a^{(i)}, x_p^{(i)}, x_n^{(i)})$ form a triangle $\triangle x_a^{(i)} x_p^{(i)} x_n^{(i)}$ whose sides are denoted as s_{ap} , s_{an} , and s_{pn} . The original triplet constraint equation (1) enforces that s_{an} is longer than s_{ap} . Because the anchor and positive samples are of the same class, a symmetrical triplet constraint can be derived, which enforces $D(h_a^{(i)}, h_p^{(i)}) + m \leq D(h_p^{(i)}, h_n^{(i)})$. According to the cosine formula, it can be proved that the angle $\angle x_a^{(i)} x_p^{(i)} x_n^{(i)}$ surrounded by the longer edges s_{an} and s_{pn} has to be the smallest one; that is, $\angle x_a^{(i)} x_p^{(i)} x_n^{(i)} \leq \min(\angle x_n^{(i)} x_a^{(i)} x_p^{(i)}, \angle x_n^{(i)} x_p^{(i)} x_a^{(i)})$. Furthermore, because $\angle x_a^{(i)} x_p^{(i)} x_n^{(i)} + \angle x_n^{(i)} x_a^{(i)} x_p^{(i)} + \angle x_n^{(i)} x_p^{(i)} x_a^{(i)} = 180^\circ$, $\angle x_a^{(i)} x_p^{(i)} x_n^{(i)}$ has to be less than 60° , which can be used to constrain the upper bound $\angle x_a^{(i)} x_p^{(i)} x_n^{(i)}$ for each triplet triangle [25]:

$$\angle x_a^{(i)} x_p^{(i)} x_n^{(i)} \leq \alpha, \quad (5)$$

where α is a predefined parameter. However, a straightforward implementation of equation (5) becomes unstable in some special cases [25]. In order to solve this problem, the angular loss function is defined in the following equation:

$$L_{\text{AL}} = \frac{1}{N^{\text{AL}}} \sum_{i=1}^{N^{\text{AL}}} \max\left(0, D(h_a^{(i)}, h_p^{(i)}) - 4D\left(h_n^{(i)}, \frac{h_a^{(i)} + h_p^{(i)}}{2}\right) \tan^2 \alpha\right), \quad (6)$$

where N^{AL} is the number of triplets, which is also equal to the batch size M in our implementation.

3.1.4. Triplet Center Loss. The goal of TCL is to leverage the advantages of the triplet loss and the center loss [39], that is, to effectively decrease the intraclass distances and increase the interclass distances simultaneously. Let a given training batch $(x^i, y^i)_{i=1}^M$ consist of M samples x^i with the associated labels y^i . In TCL, it is assumed that the features of each class share one center. Thus, the TCL is defined as

$$L_{\text{TCL}} = \frac{1}{M} \sum_{i=1}^M \max(0, m + D(h_a^{(i)}, c_{y^i}) - D(h_a^{(i)}, c_n)), \quad (7)$$

where c_{y^i} is the center of class y^i and c_n is the negative center nearest to c_{y^i} .

3.1.5. Quadruplet Center Loss. In equation (7), TCL aims to make the distance between the anchor and its corresponding center c_{y^i} smaller than the distance between the anchor and its nearest negative center c_n . For example, in Figure 3(a), the anchor can be pushed to its center. However, in some cases such as the one in Figure 3(b), it cannot, even though it is still

FIGURE 3: Two cases in TCL. Minimizing TCL can push the anchor to its center in (a) but it cannot in (b).

far away from its center. In order to solve this problem, we propose a quadruplet center loss:

$$L_{\text{QCL}} = \frac{1}{M} \sum_{i=1}^M \max \left(0, m + D(h_a^{(i)}, c_{y^i}) - \min_{(c_1, c_2)} \{D(c_1, c_2)\} \right), \quad (8)$$

where (c_1, c_2) is a pair of centers. Thus, the proposed loss function guarantees that an anchor that is not close enough to its center will be moved closer, while an anchor that is already close enough to its center will be neglected.

3.2. Search Space. Our key insight of constructing a search space and finding an optimal loss from it automatically lies in that (1) designing a loss architecture requires a lot of expert knowledge; (2) the available losses are all task- and dataset-dependent; (3) there is no rule of thumb to choose the hyperparameters for each loss.

Therefore, we use the five loss architectures to construct the search space. Each loss comes with a magnitude. We discretize the range of the magnitudes into 11 values (uniform spacing) from 0.0 to 1.0 with a step of 0.1 so that we can use a discrete search algorithm to find them. Similarly, we also discretize the margin of each loss having a margin into 51 values (uniform spacing) from 0.0 to 5.0 with a step of 0.1. For the angular loss, we discretize the angle into 51 values (uniform spacing) from 10.0° to 60.0° with a step of 1.0° . Finding an optimal loss architecture now becomes a search problem in a space of $(11 \times 51)^5 \approx 5.6 \times 10^{13}$ possibilities. Notice that there is no explicit *discard* action in our search space; this action is implicit and can be achieved by making the magnitude of a subloss architecture be 0.

3.3. Search Algorithm. The search algorithm we use to find the optimal loss architecture is a reinforcement learning algorithm (specifically, the proximal policy optimization (PPO)), inspired by [4, 27]. The loss generator is a one-layer LSTM with 100 hidden units. As shown in Figure 4, at each step, the generator takes an action according to the maximum probability produced by a softmax; the output is then fed into the next step. In total, the generator has 10 softmax layers in order to predict 5 subloss architectures, each with 2 actions (selecting a magnitude and a margin/angle).

As shown in Figure 2, the generator is trained with a reward signal, which measures how good the loss

architecture is at improving the accuracy of the object embedding network. After generating a loss architecture L with probability P , the embedding network is trained on the training set and fine-tuned using the few-shot instances. It is then evaluated on the validation set to measure the validation accuracy (the reward signal). At the end of the search, we obtain the best loss architecture L^* with the highest accuracy on the validation set, which is then used to optimize the embedding network finally.

4. Experiments and Results

To evaluate the effectiveness of the proposed approach, we conduct experiments on three popular datasets for few-shot learning, CIFAR-100 [40], Omniglot [41], and mini-ImageNet [9]. Omniglot dataset is the most popular few-shot object recognition benchmark with handwritten characters, and mini-ImageNet dataset is a subset of ImageNet [5] released recently.

4.1. Datasets

4.1.1. CIFAR-100 Dataset. The CIFAR-100 dataset [40] contains 60,000 32×32 images with 100 classes. We use 60, 20, and 20 classes for training, validation, and testing, respectively. The validation set is used to generate validation accuracies as the reward signals for LAS.

4.1.2. Omniglot Dataset. This dataset contains 32,460 images of handwritten characters, with 1,623 different characters within 50 alphabets. We follow the most common split in [8], splitting the dataset into a background set of 1200 classes and a testing set of 423 classes. We further split the background set into a training set of 800 classes and a validation set of 400 classes.

4.1.3. mini-ImageNet Dataset. The mini-ImageNet dataset is proposed by [9] as a benchmark with images of much higher resolutions and complexity. This dataset contains 100 classes randomly sampled from the ImageNet dataset, and each class contains 600 images. It is further split into a training set of 64 classes, a validation set of 16 classes, and a testing set of 20 classes [9].

FIGURE 4: Recursively constructing loss architectures by the generator.

4.2. Experimental Settings

4.2.1. Evaluation Setting. Typically, a few-shot learning task with N new classes and k instances per class is referred to as an N -way k -shot task. In this paper, all experiments are for N -way k -shot object recognition, and the evaluation setting is similar to those in other compared methods. For each iteration in LAS (see the left part of Figure 2), we train the embedding network with the generated loss L using the training set. For evaluating L , we randomly select the k -shot instances as a support set consisting of N classes with k -labeled images per class from the validation set for fine-tuning the embedding network. Then, we randomly select 15 images (disjoint with the support set) within the selected N classes from the validation set to evaluate the fine-tuned embedding network by 1-Nearest Neighbor (1NN). We repeat such evaluation procedures 5 times for each support set and use the mean accuracy as the final validation accuracy (i.e., the reward signal) corresponding to L .

After obtaining the final loss architecture L^* for one dataset, we use the testing procedure to obtain the final accuracy (see the right part of Figure 2). We first train the embedding network with L^* using the same training set. In the testing stage, we fine-tune the embedding network using a randomly selected support set from the testing set. After that, we randomly select 15 images (disjoint with the support set) within the selected N classes and then measure the classification accuracy by 1NN. To make the accuracy more convincing, we repeat such a testing procedure 600 times and report the final mean accuracy as well as the 95% Confidence Intervals (CIs).

4.2.2. Network Architecture and Parameter Setting. Our embedding network follows the same architecture as that used in [8, 9], which has four convolutional layers (Conv-4). Each convolutional layer is designed with 3×3 convolution with 64 filters followed by batch normalization, ReLU nonlinearity, and 2×2 max pooling.

On each dataset, the generator samples 5,000 loss architectures. We follow the training procedure and hyperparameters from [4] for training the generator. For training the embedding network, we use Adam [42] to perform stochastic optimization over the learning objective. The hyperparameters of Adam (i.e., β_1 , β_2 , and ϵ) are set to 0.9, 0.999, and 10^{-8} , respectively.

4.3. Ablation Study. Since LAS integrates the five sublosses into the reinforcement learning framework, we conduct an ablation study on CIFAR-100 to understand the contribution of each subloss. In each case, we remove one subloss from the search space and perform the whole process of loss architecture search as shown in Figure 2. The results are presented in Table 1.

From Table 1, we can see that removing any one subloss results in an obvious performance drop compared with using all of them. This implies that the five losses are not redundant. In addition, removing QCL, the loss proposed in this paper, causes the most significant performance reduction on this dataset, which shows that QCL may contribute the most among the five losses.

4.4. Compared Methods. To verify the performance of our LAS for few-shot learning, we compare it with the following 19 state-of-the-art methods: (1) Siamese Network (SN) [21], which presents a strategy for performing few-shot classification by learning a deep convolutional embedding network with a pairwise Siamese loss, (2) Matching Network (MN) [8], which performs few-shot learning by embedding a small labeled support set and unlabeled examples, (3) Matching Network with Full Contextual Embeddings (MN-FCE) [8], which is an upgraded version of MN by utilizing a bi-LSTM to contextually embed samples, (4) Model Agnostic Metalearning (MAML) [29], which learns hyperparameters through gradient descent in a metalearning fashion, (5) Metalearner LSTM (ML-LSTM) [9], which uses a LSTM as a metalearner to learn the parameter update rule for

TABLE 1: Mean accuracy (%) $\pm$ CIs (%) of our LAS method in six cases on CIFAR-100 dataset.

Method	5-way accuracy	
	1-Shot	5-Shot
LAS (no HMTL)	55.23 $\pm$ 0.78	70.93 $\pm$ 0.74
LAS (no MSML)	55.14 $\pm$ 0.81	70.89 $\pm$ 0.75
LAS (no AL)	55.21 $\pm$ 0.73	69.81 $\pm$ 0.81
LAS (no TCL)	54.63 $\pm$ 0.79	69.88 $\pm$ 0.76
LAS (no QCL)	54.56 $\pm$ 0.83	69.50 $\pm$ 0.83
LAS	55.85 $\pm$ 0.76	71.91 $\pm$ 0.73

In each of the first five cases, one subloss is removed from the search space. In the last case, all the sublosses are used.

optimizing the network, (6) Siamese with Memory (SM) [31], which proposes a large-scale life-long memory module to remember past training samples and makes predictions based on stored previous samples, (7) Metanetwork (Meta-N) [30], which learns metalevel knowledge across tasks for rapid generalization, (8) Meta-Stochastic Gradient Descent (Meta-SGD) [43], which learns the initialization, update direction, and learning rates by metalearning, (9) Prototypical Networks (PN) [10], which learn a prototype representation for each class, (10) Relation Network (RN) [20], which computes relation scores between few-shot instances and testing instances, (11) Memory Matching Network (MM-Net) [32], which augments CNNs with memory and learns the network parameters for unlabeled images, (12) MM-Net⁻ [32], which is a variant of MM-Net by using a mixed training strategy, (13) Graph Neural Network (GNN) [44], which casts few-shot learning as a message passing task, (14) Task Agnostic Metalearning (TAML) [33], which learns an unbiased initial model with the largest uncertainty, (15) task-dependent adaptive metric (TADAM) [45], which proposes a practical end-to-end optimization procedure to learn a task-dependent metric space, (16) Simple Neural Attentive Learner (SNAIL) [46], which proposes a generic metalearner architecture for few-shot learning, (17) Deep Nearest Neighbor Neural Network (DN4) [14], which replaces an image-level feature-based measure with a local feature-based image-to-class measure, and (18) deep learning with knowledge transfer architecture (KTN) [47], which jointly incorporates classifier learning, knowledge inferring, and visual feature learning into one framework.

Note that, for fair comparison, the backbones of LAS and all these compared methods are of the same architecture as that used in [8, 9], which is a shallow network with only 4 convolutional layers.

4.5. Results on CIFAR-100. Table 2 compares LAS with 4 state-of-the-art methods (MN, SM, MAML, and Meta-N) and the 5 sublosses described in the last section. The codes of these 4 methods are available publicly and thus can be used to run this experiment. The results across the 5-way 1-shot task and the 5-way 5-shot task show that our LAS method obtains the best performances against these methods. In particular, the 1-shot and 5-shot accuracy of our method can achieve 55.85% and 71.91%, respectively, on 5-way learning,

TABLE 2: Mean accuracy (%) $\pm$ CIs (%) of 4 state-of-the-art methods, the 5 sublosses, and LAS on the CIFAR-100 dataset.

Method	5-way accuracy	
	1-Shot	5-Shot
MN	50.90 $\pm$ 1.07	60.75 $\pm$ 0.94
SM	48.56 $\pm$ 0.87	57.07 $\pm$ 0.86
MAML	49.29 $\pm$ 0.89	66.04 $\pm$ 0.94
Meta-N	49.84 $\pm$ 0.90	67.08 $\pm$ 0.90
HMTL	49.20 $\pm$ 0.82	66.49 $\pm$ 0.99
MSML	49.73 $\pm$ 0.84	67.48 $\pm$ 0.86
AL	51.34 $\pm$ 0.74	68.84 $\pm$ 0.85
TCL	51.88 $\pm$ 0.97	68.60 $\pm$ 0.81
QCL	52.09 $\pm$ 0.78	68.97 $\pm$ 0.78
LAS	55.85 $\pm$ 0.76	71.91 $\pm$ 0.73

making the absolute improvement over MAML by 6.56% and 5.87%.

Table 2 also shows the performances of 5 metric learning approaches for few-shot object recognition (HMTL [23], MSML [24], AL [25], TCL [26], and QCL). The results indicate that no single subloss dominates the loss searched by LAS.

4.6. Results on Omniglot. Table 3 compares LAS with 8 state-of-the-art methods on the Omniglot dataset. The results of these methods are from [14, 32, 33]. The results across the 20-way 1-shot task and the 20-way 5-shot task indicate that our LAS method again achieves the best performances against other state-of-the-art techniques including embedding models (SN, MN, SM, PN, and MM-Net) and metalearning approaches (Meta-N, MAML, and TAML). In particular, the 1-shot and 5-shot accuracy of LAS can reach 97.69% and 99.21% on 20-way learning, respectively. It makes the absolute improvement over MAML by 1.89% and 0.31%, which is significant on this dataset.

4.7. Results on mini-ImageNet. The performance comparison with 14 state-of-the-art methods on mini-ImageNet is summarized in Table 4. The results of these methods are from [14, 20, 32, 33]. Our LAS method also performs the best. In particular, the 1-shot and 5-shot accuracy can achieve 54.97% and 71.92% on 5-way learning, respectively, making the absolute improvement over MM-Net by 1.60% and 4.95%. Compared with the most recent work DN4, LAS improves the accuracy of the two tasks by 3.73% and 0.90%, respectively. Our LAS gains 4.53% and 6.60% improvements over RN. RN computes relation scores between few-shot instances and the testing instances, while LAS finds the best metric learning-based loss to generate the best embedding functions.

To evaluate the proposed method on deeper networks, several experiments on mini-ImageNet are conducted by using ResNet-12 as the backbone. ResNet-12 is a 12-layer residual network. The performance comparison with ResNet-12 as the backbone on mini-ImageNet is summarized in Table 5. The results across the 5-way

TABLE 3: Mean accuracy (%) $\pm$ CIs (%) of 8 state-of-the-art methods and LAS on the Omniglot dataset with Conv-4 as the backbone.

Method	20-way accuracy	
	1-Shot	5-Shot
SN [†]	88.2	97.0
MN [†]	93.8	98.5
SM [†]	95.0	98.6
Meta-N [†]	97.0	—
PN [†]	96.0	98.9
MAML [†]	95.8 $\pm$ 0.3	98.9 $\pm$ 0.2
TAML [*]	95.12 $\pm$ 0.39	98.58 $\pm$ 0.10
MM-Net [†]	97.16 $\pm$ 0.10	98.93 $\pm$ 0.05
LAS	97.69 $\pm$ 0.17	99.21 $\pm$ 0.09

[†]Results from [32]. ^{*}Results from [33]. [†]Results from [14].

TABLE 4: Mean accuracy (%) $\pm$ CIs (%) of 14 state-of-the-art methods and LAS on the mini-ImageNet dataset with Conv-4 as the backbone.

Method	5-way accuracy	
	1-Shot	5-Shot
MN [†]	43.40 $\pm$ 0.78	51.09 $\pm$ 0.71
MN-FCE [†]	43.56 $\pm$ 0.84	55.31 $\pm$ 0.73
ML-LSTM [†]	43.44 $\pm$ 0.77	60.60 $\pm$ 0.71
SNAIL [*]	45.10	55.20
MAML [†]	48.70 $\pm$ 1.84	63.11 $\pm$ 0.92
Meta-N [†]	49.21 $\pm$ 0.96	—
PN [†]	49.42 $\pm$ 0.78	68.20 $\pm$ 0.66
GNN [†]	49.02 $\pm$ 0.98	63.50 $\pm$ 0.84
RN [‡]	50.44 $\pm$ 0.82	65.32 $\pm$ 0.70
Meta-SGD [*]	50.47 $\pm$ 1.87	64.03 $\pm$ 0.94
DN4 [†]	51.24 $\pm$ 0.74	71.02 $\pm$ 0.64
TAML [*]	51.77 $\pm$ 1.86	65.60 $\pm$ 0.93
MM-Net ^{-†}	52.74 $\pm$ 0.45	65.82 $\pm$ 0.37
MM-Net [†]	53.37 $\pm$ 0.48	66.97 $\pm$ 0.35
KTN [†]	54.61 $\pm$ 0.80	71.21 $\pm$ 0.66
LAS	55.14 $\pm$ 0.46	71.92 $\pm$ 0.48

[†]Results from [32]. [‡]Results from [20]. ^{*}Results from [33]. [†]Results from [14].

1-shot task and the 5-way 5-shot task show that the proposed method achieves the best performances against other state-of-the-art techniques. In particular, the 1-shot and 5-shot accuracy of LAS with ResNet-12 as the backbone can reach 59.44% and 77.82% on 5-way learning, respectively. It makes the absolute improvement over MAML by 9.08% and 10.48%, which is significant on this dataset.

4.8. Training and Inference Time. Similar to other reinforcement learning algorithms, the training of LAS is time-consuming. On Nvidia Tesla P100 GPU $\times$ 2, LAS takes about two days to find the best loss architecture on the mini-ImageNet dataset for the 5-way 5-shot task. However, after training, its inference is fast enough; for example, it only takes 2 ms to perform one inference for the same task.

TABLE 5: Mean accuracy (%) $\pm$ CIs (%) of 7 state-of-the-art methods and LAS on the mini-ImageNet dataset with ResNet-12 as the backbone.

Method	5-way accuracy	
	1-Shot	5-Shot
MN	45.65 $\pm$ 0.86	53.64 $\pm$ 0.81
MAML	50.36 $\pm$ 0.78	67.34 $\pm$ 0.93
Meta-N	52.07 $\pm$ 0.89	68.84 $\pm$ 0.87
Meta-SGD	52.96 $\pm$ 0.75	69.52 $\pm$ 0.71
SNAIL [†]	55.71 $\pm$ 0.99	68.88 $\pm$ 0.92
MetaNet [†]	57.10 $\pm$ 0.70	70.04 $\pm$ 0.63
TADAM [†]	58.50 $\pm$ 0.30	76.70 $\pm$ 0.30
LAS	59.44 $\pm$ 0.41	77.82 $\pm$ 0.34

[†]Results from [48].

5. Conclusion

In this paper, we have proposed an automatic approach to best loss architecture search for few-shot object recognition. Five metric learning-based sublosses (with one developed by us) are used to construct the search space. The loss generator is trained by a reinforcement learning algorithm. Our experiments show that the proposed few-shot object recognition method outperforms other state-of-the-art methods on three popular benchmarks. The future work includes combining network architecture search and loss architecture search for better AutoML.

Data Availability

The datasets and source codes used to support the findings of this study are available from the author upon request via e-mail (jyue@pku.edu.cn).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 41701500, in part by the Natural Science Foundation of Hunan Province under Grants 2018JJ3641 and 2019JJ60001, and in part by Innovation-Driven Project of Central South University under Grant 2020CX036.

References

- [1] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proceedings of the ICLR*, San Diego, CA, USA, May 2015.
- [2] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the ICLR*, San Diego, CA, USA, May 2015.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the ICLR*, San Juan, Puerto Rico, May 2016.

- [4] B. Zoph, V. Vasudevan, J. Shlens, V. Quoc, and Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the ICLR*, Vancouver, BC, Canada, April 2018.
- [5] O. Russakovsky, D. Jia, H. Su et al., "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, 2015.
- [6] A. Kuznetsova, R. Hassan, A. Neil et al., "The open images dataset v4," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, 2020.
- [7] R. Ali Sharif, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the CVPR Workshops*, Boston, MA, USA, April 2014.
- [8] Oriol Vinyals, C. Blundell, T. Lillicrap, D. Wierstra et al., "Matching networks for one shot learning," in *Proceedings of the NeurIPS*, Barcelona, Spain, December 2016.
- [9] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proceedings of the ICLR*, San Juan, Puerto Rico, May 2016.
- [10] Jake Snell, S. Kevin, and R. Zemel, "Prototypical networks for few-shot learning," in *Proceedings of the NeurIPS*, Barcelona, Spain, December 2017.
- [11] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proceedings of the ICCV*, Venice, Italy, October 2017.
- [12] E. Schwartz, L. Karlinsky, S. Joseph et al., "Delta-encoder: an effective sample synthesis method for few-shot object recognition," in *Proceedings of the NeurIPS*, Montreal, Canada, December 2018.
- [13] S. Krishnagopal, Y. Aloimonos, and M. Girvan, "Similarity learning and generalization with limited data: a reservoir computing approach," *Complexity*, vol. 2018, pp. 1–15, Article ID 6953836, 2018.
- [14] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *Proceedings of the CVPR*, Long Beach, MA, USA, June 2019.
- [15] X. Dong, L. Zheng, Ma Fan, Yi Yang, and D. Meng, "Few-example object detection with model communication," in *Proceedings of the IEEE TPAMI*, Piscataway, NJ, USA, June 2018.
- [16] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the CVPR*, Long Beach, MA, USA, June 2017.
- [17] X. Dong, L. Zhu, De Zhang, Yi Yang, and F. Wu, "Fast parameter adaptation for few-shot image captioning and visual question answering," in *Proceedings of the ACM MM*, Seoul, Republic of Korea, May 2018.
- [18] J. Choe, S. Park, K. Kim, J. H Park, D. Kim, and H. Shim, "Face generation for low-shot learning using generative adversarial networks," *ICCV*, 2017.
- [19] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Yi Yang, "Exploit the unknown gradually: one-shot video-based person re-identification by stepwise learning," in *Proceedings of the CVPR*, Long Beach, MA, USA, June 2018.
- [20] F. Sung, Y. Yang, Li Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: relation network for few-shot learning," in *Proceedings of the CVPR*, Long Beach, MA, USA, June 2018.
- [21] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proceedings of the ICML Workshops*, Lille, France, July 2015.
- [22] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *Proceedings of the CVPR*, Long Beach, MA, USA, June 2017.
- [23] F. Schroff, D. Kalenichenko, and P. James, "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the CVPR*, Columbus, OH, USA, June 2015.
- [24] Q. Xiao, H. Luo, and C. Zhang, "Margin sample mining loss: a deep learning based method for person re-identification," 2017, <http://arxiv.org/abs/1710.00478>.
- [25] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *Proceedings of the ICCV*, Long Beach, CA, USA, April 2017.
- [26] X. He, Y. Zhou, Z. Zhou, B. Song, and B. Xiang, "Triplet-center loss for multi-view 3d object retrieval," in *Proceedings of the CVPR*, Long Beach, MA, USA, June 2018.
- [27] B. Zoph and V. Quoc, "Neural architecture search with reinforcement learning," in *Proceedings of the ICLR*, Toulon, France, April 2017.
- [28] John Schulman, W. Filip, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, <http://arxiv.org/abs/1707.06347>.
- [29] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the ICML*, Sydney, Australia, August 2017.
- [30] T. Munkhdalai and H. Yu, "Meta networks," *Proceedings of Machine Learning Research*, vol. 70, pp. 2554–2563, 2017.
- [31] Ł. Kaiser, O. Nachum, A. Roy, and S. Bengio, "Learning to remember rare events," in *Proceedings of the ICLR*, Toulon, France, April 2017.
- [32] C. Qi, Y. Pan, T. Yao, C. Yan, and T. Mei, "Memory matching networks for one-shot image recognition," in *Proceedings of the CVPR*, Long Beach, MA, USA, June 2018.
- [33] M. Abdullah Jamal and G.-J. Qi, "Task agnostic meta-learning for few-shot learning," in *Proceedings of the CVPR*, Long Beach, MA, USA, June 2017.
- [34] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the CVPR*, New York, NY, USA, June 2006.
- [35] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proceedings of the ECCV*, Amsterdam, The Netherlands, October 2016.
- [36] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the CVPR*, New York, NY, USA, June 2006.
- [37] J. Wang, Y. Song, T. Leung et al., "Learning fine-grained image similarity with deep ranking," in *Proceedings of the CVPR*, Columbus, OH, USA, June 2014.
- [38] A. Li, T. Luo, T. Xiang, W. Huang, and L. Wang, "Few-shot learning with global class representations," in *Proceedings of the ICCV*, Long Beach, CA, USA, April 2019.
- [39] Y. Wen, K. Zhang, Z. Li, and Yu Qiao, "A discriminative feature learning approach for deep face recognition," in *Proceedings of the ECCV*, Amsterdam, The Netherlands, October 2016.
- [40] A. Krizhevsky and G. Hinton, *Learning Multiple Layers of Features from Tiny Images*, Technical report, Citeseer, New York, NY, USA, 2009.
- [41] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, Boston, MA, USA, June 2011.
- [42] D. P. Kingma and Ba Jimmy, "Adam: a method for stochastic optimization," in *Proceedings of the ICLR*, San Diego, CA, USA, May 2015.
- [43] Z. Li, F. Zhou, F. Chen, and H. Li, *Meta-sgd: learning to learn quickly for few-shot learning*, 2017, <http://arxiv.org/abs/1707.09835>.

- [44] V. Garcia Satorras and E. Joan Bruna, “Few-shot learning with graph neural networks,” in *Proceedings of the ICLR*, Vancouver, BC, Canada, April 2018.
- [45] B. Oreshkin, L. Pau Rodríguez, and A. Lacoste, “Tadam: task dependent adaptive metric for improved few-shot learning,” in *Proceedings of the NIPS*, Montreal, Canada, December 2018.
- [46] N. Mishra, M. Rohaninejad, Xi Chen, and P. Abbeel, “A simple neural attentive meta-learner,” in *Proceedings of the CVPR*, Long Beach, CA, USA, June 2018.
- [47] Z. Peng, Z. Li, J. Zhang, Y. Li, G.-J. Qi, and J. Tang, “Few-shot image recognition with knowledge transfer,” in *Proceedings of the ICCV*, Long Beach, CA, USA, April 2019.
- [48] S. Gidaris and N. Komodakis, “Generating classification weights with gnn denoising autoencoders for few-shot learning,” in *Proceedings of the CVPR*, Long Beach, CA, USA, June 2019.