

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

6-2022

Revisiting local descriptor for improved few-shot classification

Jun HE

Richang HONG

Xueliang LIU

Mingliang XU

Qianru SUN

Singapore Management University, qianrusun@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Databases and Information Systems Commons](#), and the [Graphics and Human Computer Interfaces Commons](#)

Citation

HE, Jun; HONG, Richang; LIU, Xueliang; XU, Mingliang; and SUN, Qianru. Revisiting local descriptor for improved few-shot classification. (2022). *ACM Transactions on Multimedia Computing, Communications and Applications*. 18, (2s), 1-23.

Available at: https://ink.library.smu.edu.sg/sis_research/7558

This Journal Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Revisiting Local Descriptor for Improved Few-Shot Classification

JUN HE, RICHANG HONG*, and XUELIANG LIU, Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, China and Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, China

MINGLIANG XU, Zhengzhou University, China

QIANRU SUN, Singapore Management University, Singapore

Few-shot classification studies the problem of quickly adapting a deep learner to understanding novel classes based on few support images. In this context, recent research efforts have been aimed at designing more and more complex classifiers that measure similarities between query and support images, but left the importance of feature embeddings seldom explored. We show that the reliance on sophisticated classifiers is not necessary, and a simple classifier applied directly to improved feature embeddings can instead outperform most of the leading methods in the literature. To this end, we present a new method named **DCAP** for few-shot classification, in which we investigate how one can improve the quality of embeddings by leveraging **Dense Classification and Attentive Pooling**. Specifically, we propose to train a learner on base classes with abundant samples to solve dense classification problem first and then meta-train the learner on a bunch of randomly sampled few-shot tasks to adapt it to few-shot scenario or the test time scenario. During meta-training, we suggest to pool feature maps by applying attentive pooling instead of the widely used global average pooling (GAP) to prepare embeddings for few-shot classification. Attentive pooling learns to reweight local descriptors, explaining what the learner is looking for as evidence for decision making. Experiments on two benchmark datasets show the proposed method to be superior in multiple few-shot settings while being simpler and more explainable. Code would be made public at: <https://github.com/Ukeyboard/dcap/>.

CCS Concepts: • **Computing methodologies** → **Image representations; Object recognition; Supervised learning by classification; Transfer learning; Neural networks; Feature selection.**

Additional Key Words and Phrases: few-shot learning, image classification, visual recognition, meta-learning, attention networks

ACM Reference Format:

Jun He, Richang Hong, Xueliang Liu, Mingliang Xu, and Qianru Sun. 2022. Revisiting Local Descriptor for Improved Few-Shot Classification. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1, Article 1 (January 2022), 23 pages. <https://doi.org/10.1145/3511917>

*Corresponding author.

Authors' addresses: Jun He, hyj.hfut.mail@gmail.com; Richang Hong, hongrc.hfut@gmail.com; Xueliang Liu, liuxueliang1982@gmail.com, Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, 193 Tunxi Rd, Hefei, Anhui, China, 230009 and Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, 5089 Wangjiang West Rd, Hefei, Anhui, China, 230026; Mingliang Xu, iexumingliang@zzu.edu.cn, Zhengzhou University, 100 Science Ave, Zhengzhou, Henan, China, 450001; Qianru Sun, qianrusun@smu.edu.sg, Singapore Management University, 80 Stamford Rd, Singapore, 178902.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1551-6857/2022/1-ART1 \$15.00
<https://doi.org/10.1145/3511917>

1 INTRODUCTION

Deep neural networks have achieved remarkable success in a variety of fields, such as object detection [4, 35], segmentation [18, 76], action recognition [52, 61], image captioning [3, 78], cross-modal retrieval [71, 72] and especially image classification [22, 39, 59, 60], since AlexNet [26] won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. However, the success hinges on the ability to apply gradient-based optimization routines to high-capacity models given hundreds if not thousands of training examples as supervision [49]. If the training data is scarce, studies [11, 65] have shown that deep learning models would fail to learn. In contrast, humans can learn fast even when taught by shown a few examples. To give deep learning models the generalization ability of humans, recently there has been an increasing interest in few-shot classification [11, 17, 21, 24, 37, 40, 42, 44, 48, 53, 69, 77].

Few-shot classification aims to tackle the problem of learning a novel set of classes based on few support images. Generally, given a learner that is conceptually composed of an embedding network and a classifier, one straightforward approach towards this problem is transfer learning, *i.e.* fine-tuning a pre-trained embedding network, as well as a newly initialized classifier, on the few support images so as to adapt them to the target recognition task, since transfer learning is proven effective in conventional supervised learning scenarios. Whereas, in few-shot setting, the naive learning procedure is at high risk of overfitting due to the lack of training examples. Data augmentation or regularization techniques [16, 66] may alleviate overfitting by increasing data diversity or shrinking model searching space, yet they can not solve it. To tackle the challenge, Vinyals *et al.* [65] introduced an innovative learning paradigm called meta-learning, by following the idea of “learning to learn”. Instead of directly optimizing the learner upon the few support images, meta-learning assumes the existence of a set of base classes with sufficient examples per class, divided into disjoint meta-training, meta-validation and meta-testing splits, and proposes to train the learner on a collection of few-shot tasks randomly sampled from the meta-training split (also referred as meta-training set), targeting at learning transferable knowledge from these similar tasks to facilitate the learning of novel classes when applied to a new task. Effective and elegant, meta-learning has become the *de facto* solution to few-shot classification.

While more and more meta-learning approaches are proposed, we notice that most of them concentrate on reformulating the classifier that is responsible for measuring the similarities between query and support images, but leave the feature embeddings that are used for classification seldom explored. For example, Vinyals *et al.* [65] proposed an attention-based classifier by comparing a query against all support images. Snell *et al.* [54] computed the squared Euclidean distances from the query to all category prototypes for classification. Sung *et al.* [58] and He *et al.* [17] formulated the classifier as a relation network that learns to estimate the similarity of a query-support pair. Zhang *et al.* [77] in addition proposed an uncertainty-aware classifier, by taking observation noises in the query-support similarities into account. As for the feature embeddings, these methods typically apply global average pooling (GAP) over feature maps to simply prepare embeddings for the classifier that follows. We argue that the undue emphasis on the classifier in previous work overly undervalues the importance of feature embeddings. Thus, in an orthogonal direction, we propose to tackle few-shot classification differently from the perspective of acquiring more discriminative and robust embeddings in this work. To be specific, we follow a two-stage training paradigm for model optimization and representation learning, namely pre-training and meta-finetuning. Whenever a set of novel classes with few support images and a set of base classes with sufficient examples are provided, our first step is to train the learner on the base classes to perform large-scale classification with standard cross-entropy loss. After that, we fine-tune the learner on randomly sampled few-shot tasks in the meta-training set to adapt it to the few-shot

scenario or the test time scenario. Without bells and whistles, we demonstrate that the two-stage training is able to achieve good results and, especially, the pre-training can bring more than 4% improvement on average over models meta-trained from scratch, which indicates the effectiveness of good embeddings.

In addition to the two-stage training, we further propose to leverage dense classification and attentive pooling to improve the quality of feature embeddings. Particularly, we propose to perform dense classification over local descriptors in the pre-training stage. In meta-finetuning, we develop an attentive pooling strategy for summarizing local descriptors into a compact image-level embedding for few-shot classification. The proposition of dense classification is based on the observation that local descriptors within the same feature map generated by a GAP-pretrained model¹ often exhibit severe semantic discrepancy² (see Fig 1). The discrepancy is somewhat counterintuitive because convolutional layers are technically locally connected layers and each local descriptor in the final feature map thus generally has a large enough receptive field that probably covers the same semantic region of the input image. We show that dense classification in a DC-pretrained model³ can eliminate the discrepancy by enforcing semantic consistency among local descriptors. The idea of attentive pooling, on the other hand, is inspired by two insights. First, not all local image regions are equally representative to describe the target object in image. For example, as illustrated in Fig 1, the target object could only occupy a small proportion of the image, or even be contaminated by distractors. Second, the GAP strategy used in existing work does not deal with spatial information and forces to attach equal importance to each local descriptor, which not only causes information loss but also hinders the learner from attending to informative image regions, affecting the classifier negatively. By attentive pooling, we aim to select more informative local descriptors to build better image embeddings for the good of performance. We find that such a pooling strategy can consistently outperform its GAP counterparts.

We dub our method DCAP, *i.e.* Dense Classification pre-training plus meta-finetuning with Attentive Pooling for few-shot classification. Our contributions are as follows:

- Firstly, we formulate meta-learning as a two-stage training paradigm, where we introduce a dense classification pre-training stage to reduce semantic discrepancy among local descriptors and devise an attentive pooling strategy in meta-finetuning to select more informative local descriptors for few-shot classification.
- Secondly, we justify the importance of pre-training in the acquisition of strong representation ability and validate the effectiveness of meta-finetuning in the acquisition of fast generalization ability. We show that (i) pre-training can bring more than 4% improvement on average over models meta-trained from scratch, but (ii) a pre-trained model in inference mode performs no better than random guessing unless an extra meta-finetuning is carried out.
- Last but not least, we demonstrate that the reliance on sophisticated classifier in prior art is not necessary and set up a new baseline where a simple nearest-centroid classifier outperforms most of the leading methods by a large margin. On basis of empirical evaluation on *miniImagenet* [65] and *tieredImagenet* [45] datasets, we provide insights for best practices in implementation.

¹A GAP-pretrained model refers to the model in which feature maps are pooled into feature embeddings by applying GAP before classification in the pre-training stage.

²By semantic discrepancy, we mean two local descriptors have a small cosine similarity score (*e.g.* 0.1) but they ought to have a big one (*e.g.* 1.0).

³A DC-pretrained model refers to the model pre-trained with dense classification over local descriptors enabled in the pre-training stage.

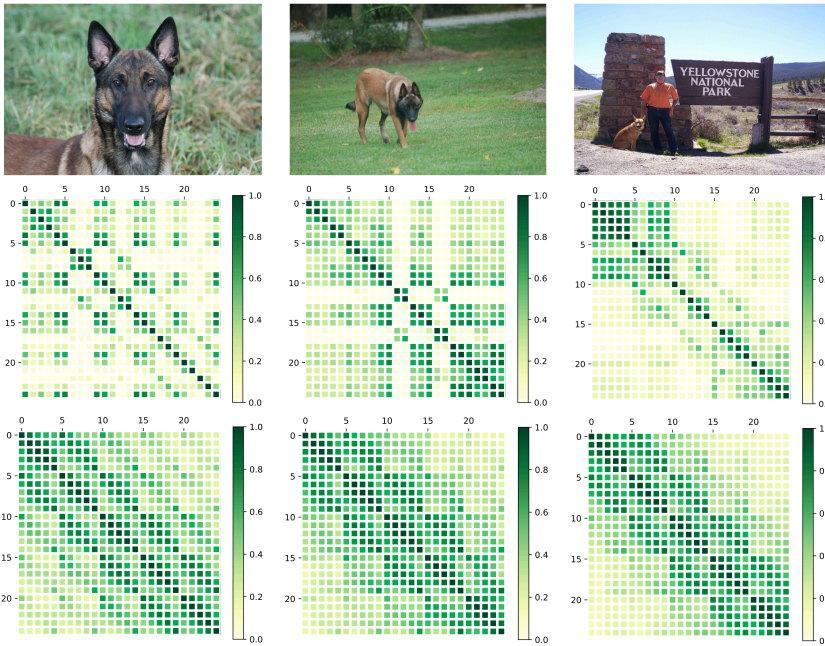


Fig. 1. **Cosine similarities between local descriptors.** From top to bottom: sample images, similarities between local descriptors in the feature map generated by a GAP-pretrained model, and similarities between local descriptors in the feature map generated by a DC-pretrained model. From left to right: salient target object in the middle (ideal example for training and testing), small target object in clear background, and small target object contaminated by distractors.

2 RELATED WORK

2.1 Meta-learning for Few-shot Classification

Meta-learning provides a paradigm where a learner gains experience over plenty of related but distinguished tasks and thereafter uses the experience to efficiently solve similar but previously unseen tasks [70]. This ‘learning-to-learn’ is recognized to show many benefits, including but not limited to data and compute efficiency [20]. To learn from few examples, few-shot classification thus has widely adopted this very paradigm and achieved promising results. Approaches in this line can be broadly grouped into two: optimization-based [1, 11, 28, 33, 41, 44, 48] and metric-based [17, 24, 29, 42, 54, 65]. In this part, we briefly review the two branches and cite some known work in the literature. For a complete list of recent work on meta-learning, we refer readers to [2, 20].

Optimization-based approaches intend to train a meta-learner either to learn a general optimization algorithm as an alternative of the SGD optimizer for updating the target learner during training, or to learn a good initialization of the learner that allows for quick adaptation to a new task in a small number of model updating with a small amount of training data. Representative work in this branch include Meta-learner LSTM [44], MAML [11] and the variants of MAML, such as Reptile [41], LEO [48], MetOptNet [28] and ALFA [1], to name a few. In Meta-learner LSTM, the authors described an LSTM-based model that is trained to discover both a task-common initialization for the learner’s parameters, as well as an update rule for updating the learner’s parameters. MAML proposed a tiered meta-learning paradigm in which a good initialization of the learner’s parameters is slowly extracted in its outer-loop training and a fast adaptation within

several gradient steps is explicitly enforced in the inner-loop fine-tuning. LEO recognized the difficulties of directly operating on high-dimensional parameters in few-shot setting. Consequently, it learned a low-dimensional latent space in which optimization-based meta-learning is performed and from which the high-dimensional parameters can be generated. While optimization-based approaches have advanced few-shot classification a lot, the efforts they have made are largely made to advance the meta-learning algorithm [1, 33, 41], with little attention being paid to feature embeddings. Lately, Raghu *et al.* [43] investigated the effectiveness of MAML and suggested that the performance of optimization-based meta-learning is mainly determined by the quality of the learned embeddings. This finding is very coherent with our emphasis on the importance of high-quality feature embeddings.

Metric-based approaches normally train the learner to learn an easily transferable embedding space, where ideally datapoints in the same class keep close to each other and datapoints from different classes stay far apart. The core idea is related to nearest neighbor algorithms and deep metric learning [36, 73]. In general, metric-based approaches would transform input data into lower dimensional embeddings first and then use the embeddings to design proper classifier for clustering or measuring their similarities. For example, Vinyals *et al.* [65] presented Matching Networks which employs an attention mechanism to derive a weighted k-NN classifier. Snell *et al.* [54] took the mean vector of support images belonging to the same class as prototype and introduced a nearest-centroid classifier, *i.e.* Prototypical Networks. TADAM [42] introduced metric-scaling, task conditioning and auxiliary co-training to boost the performance of Prototypical Networks. Li *et al.* [29] further introduced an adaptive margin loss to help separate samples from different classes. To overcome limitations of using pre-defined distance metrics (*e.g.* cosine distance [65] and Euclidean distance [54]) for classification, Sung *et al.* [58] and He *et al.* [17] also formulated the distance metric as a learnable network in order to learn a generic metric that better suits few-shot setting. In this work, we employ a simple nearest-centroid classifier that is similar to the one in Prototypical Networks for few-shot classification. But unlike Prototypical Networks who are famed for the proposed classifier, we focus on producing more powerful embeddings to support the classifier, by explicitly proposing to leverage dense classification and attentive pooling to improve the quality of feature embeddings.

2.2 Towards Improving Feature Embeddings

There is a family of meta-learning approaches that devote to improving feature embeddings. Satorras and Estrach [50] and Kim *et al.* [25] explored graph neural networks in the context of few-shot classification and presented graphical models that enhance embeddings by borrowing information from neighbors in a graph. Similarly, [17, 30, 46] presented graph-like transductive algorithms that take all unlabeled query images into the embedding propagation procedure so that information in the unlabeled query images can be leveraged for embedding enhancement. [16, 66] viewed the problem differently. They presented generative algorithms to generate new training examples, and proposed to learn better embeddings from more training examples with larger intra-class diversity. In [56], Su *et al.* studied the effectiveness of utilizing self-supervised learning (SSL) techniques in few-shot setting. SSL utilizes the structure information contained already in an image to facilitate the learning of representations, having been well studied in traditional unsupervised learning with large unlabeled datasets [9, 13, 27]. Pre-training on large-scale dataset is able to yield embeddings of high quality. Dhillon *et al.* [8] and Chen *et al.* [7] both showed that fine-tuning a well pre-trained embedding network can produce results rivaling far more complex algorithms in the low-data regime. Tian *et al.* [62] also demonstrated that learning supervised or self-supervised representation on a large-scale dataset, followed by learning a linear classifier on top of this representation, could outperform leading few-shot approaches.

2.3 Local Descriptors in Few-shot Classification

In existing few-shot classification approaches, feature maps are usually pooled or flattened into embedding vectors before classification. In [31], Li *et al.* proposed to leverage local descriptors for better classification. They built an image-to-class classifier in which each class is represented by all local descriptors from support images belonging to that class, and computed the image-to-class similarity score by summing the similarities between each local descriptor in a query image and its k -nearest neighbors in the local descriptor space of that class. Wu *et al.* [68] introduced a deformable embedding network to embed an image as a set of local descriptors. They in addition proposed to consider spatial relationship of semantic objects in two images when computing similarity scores for classification. Likewise, given an image represented as a set of local descriptors, [21] and [24] proposed to adaptively match two images to highlight relevant regions so as to build more discriminative embeddings. We do not follow the complex image-to-image adaptation but instead present a simple attentive pooling to highlight informative regions. The assumption is that, relevant regions in two images would stand out spontaneously if they are the key to correct classification.

2.4 Attention in Few-shot Classification

An early adoption of attention in few-shot classification is Matching Networks [65], where Vinyals *et al.* proposed to attentively propagate label information from support images to construct the logits of a query image for classification. Laterly, Wu *et al.* [68] proposed a dual correlation attention mechanism to overcome sensitivity to spatial locations of semantic objects in two images when comparing them for similarity measurement in the relation network. Hou *et al.* [21] presented a cross attention mechanism to highlight relevant regions in two images to enhance feature discriminability. Jiang *et al.* [24] developed an attention-based meta-reweighting strategy to adapt the embeddings of query images by incorporating the embeddings of support images. The image-to-class classifier proposed in [31] can also be explained as an application of attention mechanism, especially in its finding k -nearest neighbors of a query local descriptor. Other related work include graph-based methods [17, 25, 30, 37, 46, 50] and Transformer-based method [10]. Graph-based methods are intrinsically related because the message passing mechanism in graph is usually interpreted as an attention-based information propagation action. As for Transformer [64], it is a born attention-based framework.

3 METHODOLOGY

We consider a learner that adopts a nearest-centroid classifier and propose to train the learner by following the two-stage training paradigm described in §1, *i.e.* dense classification pre-training and meta-finetuning with attentive pooling. We find that the two-stage training ensures a set of high-quality embeddings, upon which even the simple classifier can achieve results that are competitive or better than most of the leading few-shot learning approaches. In this section, we start by establishing preliminaries about problem definition and meta-learning in §3.1; then we introduce the nearest-centroid classifier employed by our method in §3.2; finally, we detail the two-stage training paradigm in §3.3 and §3.4.

3.1 Few-shot Classification with Meta-learning

A formal definition of few-shot classification considers the well-organized N -way K -shot few-shot setting, where the target task $\mathcal{T} = \{\mathcal{D}^{train}, \mathcal{D}^{test}\}$ contains $N_s = N \times K$ labelled support images $\mathcal{D}^{train} = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_s}$ with $y_i^s \in C_{novel}$ and N_q unlabeled query images $\mathcal{D}^{test} = \{\mathbf{x}_i^q\}_{i=1}^{N_q}$ from the same label space. $C_{novel} = [c_1, c_2, \dots, c_N]$ is the set of novel classes to learn in the task. Assuming a learner $\mathcal{A}(\cdot; \Theta)$ that is conceptually composed of two parts: an embedding network

$f_\theta : \mathcal{X} \rightarrow \mathbb{R}^{d \times h \times w}$ and a classifier $g_\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^N$, few-shot classification aims to adapt the learner to correctly label each query image in \mathcal{D}^{test} based on the supervision provided in \mathcal{D}^{train} with the following objective

$$\hat{\Theta} = \arg \min_{\Theta} \mathcal{L}(\mathcal{D}^{test} | \mathcal{D}^{train}; \Theta) + \mathcal{R}(\Theta), \quad (1)$$

where $\Theta = (\theta, \varphi)$ is the learner's parameters, \mathcal{L} is the loss function and \mathcal{R} is the regularization term, e.g. weight decay.

Optimizing the learner directly on the few support images in \mathcal{D}^{train} is at high risk of overfitting. Meta-learning presents an efficient way for model optimization in such an extremely low-data regime. In meta-learning, we would resort to an extra dataset \mathcal{D}^{base} that has a disjoint label space C_{base} (i.e. $C_{base} \cap C_{novel} = \emptyset$) with sufficient samples per class, and train the learner on a collection of few-shot tasks randomly sampled from \mathcal{D}^{base} . Conventionally, the dataset \mathcal{D}^{base} is divided into disjoint meta-training, meta-validation and meta-testing splits. From the meta-training split, we are allowed to sample a set of training tasks $\mathcal{D}_{meta}^{train} = \{(\mathcal{D}_i^{train}, \mathcal{D}_i^{test})\}_{i=1}^I$, where each task $\mathcal{T}_i = (\mathcal{D}_i^{train}, \mathcal{D}_i^{test})$ instantiates an N -way K -shot few-shot classification problem that mimics the target task \mathcal{T} . The learner is then trained to minimize the average test error on $\mathcal{D}_{meta}^{train}$, i.e.

$$\Theta^* = \arg \min_{\Theta} \mathbb{E}_{\mathcal{T}_i \in \mathcal{D}_{meta}^{train}} [\mathcal{L}(\mathcal{D}_i^{test} | \mathcal{D}_i^{train}; \Theta)] + \mathcal{R}(\Theta). \quad (2)$$

We also randomly sample a family of few-shot tasks $\mathcal{D}_{meta}^{val}/\mathcal{D}_{meta}^{test}$ from the meta-validation/meta-testing split. \mathcal{D}_{meta}^{val} is used to monitor the learning process for model selection. And, the average test error on $\mathcal{D}_{meta}^{test}$ is taken as an indicator of how the learner generalizes to new tasks. By learning from similar tasks, meta-learning encourages the learner to learn transferable knowledge that facilitates the learning of novel classes when applied to the target task \mathcal{T} .

3.2 The Nearest-centroid Classifier

In Prototypical Networks [54], Snell *et al.* proposed a classifier that labels a query image based on its distance to each class centroid. Formally, given the task \mathcal{T} , the classifier is defined as follows

$$y_i^q = \arg \max_t \frac{e^{s(\mathbf{f}_i^q, \mathbf{c}_t)/\tau}}{\sum_{j=1}^N e^{s(\mathbf{f}_i^q, \mathbf{c}_j)/\tau}}, \quad t = 1, 2, \dots, N \quad (3)$$

where $\mathbf{f}_i^q \in \mathbb{R}^d$ denotes the feature embedding of image $\mathbf{x}_i^q \in \mathcal{D}^{test}$, $[\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N]$ are the class centroids, $s(\cdot, \cdot)$ is a similarity function and τ is the temperature hyperparameter. To clarify, \mathbf{f}_i^q is given by applying GAP over feature map $f_\theta(\mathbf{x}_i^q) \in \mathbb{R}^{d \times h \times w}$, i.e. $\mathbf{f}_i^q = \text{GAP}(f_\theta(\mathbf{x}_i^q))$. The similarity function $s(\cdot, \cdot)$ could be cosine similarity or negative Euclidean distance. As for class centroid \mathbf{c}_j , it takes the average of support images in the j -th class (i.e. \mathcal{S}_j) as defined by

$$\mathbf{c}_j = \frac{1}{|\mathcal{S}_j|} \sum_{\mathbf{x}^s \in \mathcal{S}_j} \text{GAP}(f_\theta(\mathbf{x}^s)). \quad (4)$$

Simple but effective, as can be seen in §4, Prototypical Networks provide a strong baseline for few-shot classification.

In this work, we focus on showing the importance of feature embeddings but not devote to designing as complex a classifier as the many other prior work did, so we adopt a similar nearest-centroid classifier here for few-shot classification. On top of Prototypical Networks, we make the following changes. First, we particularly propose to replace the GAP pooling with an attentive pooling strategy for summarizing local descriptors into image-level embeddings, i.e. $\mathbf{f}_i^q = \text{AttPool}(f_\theta(\mathbf{x}_i^q))$. Second, in order to reduce the labor required in hyperparameter tuning, we reformulate (3) as

follows to eradicate hyperparameter τ ,

$$y_i^q = \arg \max_t \frac{e^{\langle \mathbf{f}_i^q, \mathbf{c}_t / \|\mathbf{c}_t\|_2 \rangle}}{\sum_{j=1}^N e^{\langle \mathbf{f}_i^q, \mathbf{c}_j / \|\mathbf{c}_j\|_2 \rangle}}, \quad t = 1, 2, \dots, N \quad (5)$$

where the dot product $\langle \mathbf{f}_i^q, \mathbf{c}_j / \|\mathbf{c}_j\|_2 \rangle$ can be viewed as the projection of vector \mathbf{f}_i^q on the $\mathbf{c}_j / \|\mathbf{c}_j\|_2$ axis.

3.3 Dense Classification Pre-training

The pre-training refers to training the embedding network and a global classifier on the meta-training split for large-scale classification with standard cross-entropy loss. Let $\mathcal{D}_{base}^{train}$ be the meta-training split, prior work [7, 8, 57] train with the following objective in this stage

$$\Theta^* = \arg \min_{\Theta} \sum_{(\mathbf{x}, y_b) \in \mathcal{D}_{base}^{train}} \mathcal{L}^{ce}(\mathbf{W}^T \text{GAP}(f_{\theta}(\mathbf{x})) + \mathbf{b}, y_b) + \mathcal{R}(\Theta), \quad (6)$$

where $\Theta = (\theta, \mathbf{W}, \mathbf{b})$ denotes learnable parameters including θ of the embedding network and (\mathbf{W}, \mathbf{b}) of the global classifier. Differently, we propose to remove the GAP operator and perform dense classification over local descriptors, *i.e.*

$$\Theta^* = \arg \min_{\Theta} \sum_{(\mathbf{x}, y_b) \in \mathcal{D}_{base}^{train}} \sum_{j=1}^r \mathcal{L}^{ce}(\mathbf{W}^T f_{\theta}^j(\mathbf{x}) + \mathbf{b}, y_b) + \mathcal{R}(\Theta), \quad (7)$$

where feature map $f_{\theta}(\mathbf{x})$ is regarded as a set of $r = h \times w$ local descriptors in \mathbb{R}^d with $f_{\theta}^j(\mathbf{x})$ representing the j -th item in set. Though, from the Mathematics point of view, GAP-pretraining in (6) and DC-pretraining in (7) are almost identical, we experimentally demonstrate that GAP-pretraining would lead to severe semantic discrepancy among local descriptors (see Fig 1) when maintaining image-level semantics. DC-pretraining can alleviate the problem by forcing to classify each local descriptor correctly, which explicitly enforces semantic consistency among local descriptors. In practice, we transform the image label y_b into one-hot hard label \mathbf{y}_b and apply label smoothing [59] with smoothing parameter $\epsilon = 0.1$ to turn it into a soft label $\tilde{\mathbf{y}}_b$ for model training.

3.4 Meta-finetuning with Attentive Pooling

The pre-training in general ends up with a tiny classification loss which indicates the presence of a set of good embeddings. However, we find that the pre-trained embeddings are not well-prepared for few-shot classification unless an extra meta-finetuning is thereafter performed. By meta-finetuning, we mean to fine-tune the pre-trained embedding network on the meta-training set $\mathcal{D}_{meta}^{train}$ to perform nearest-centroid few-shot classification following the standard meta-learning paradigm described in §3.1.

In this meta-training stage, previous work usually apply GAP over feature maps to pool local descriptors in the feature map into compact image-level embeddings for few-shot classification. We argue that different local descriptors represent different image regions and, as a consequence, they should be of different importances during the pooling operation. Thus, we propose to reweight local descriptors right before summarizing them into the image-level embedding so that some informative local descriptors in a feature map will surpass the other less informative local descriptors to contribute a more discriminative feature embedding. As illustrated in Figure 2, given a training task $\mathcal{T}_i = (\mathcal{D}_i^{train}, \mathcal{D}_i^{test})$ where $\mathcal{T}_i \in \mathcal{D}_{meta}^{train}$, images are first encoded into the embedding space, each represented by a 3-d feature map in $\mathbb{R}^{d \times h \times w}$. We next summarize each feature map into an

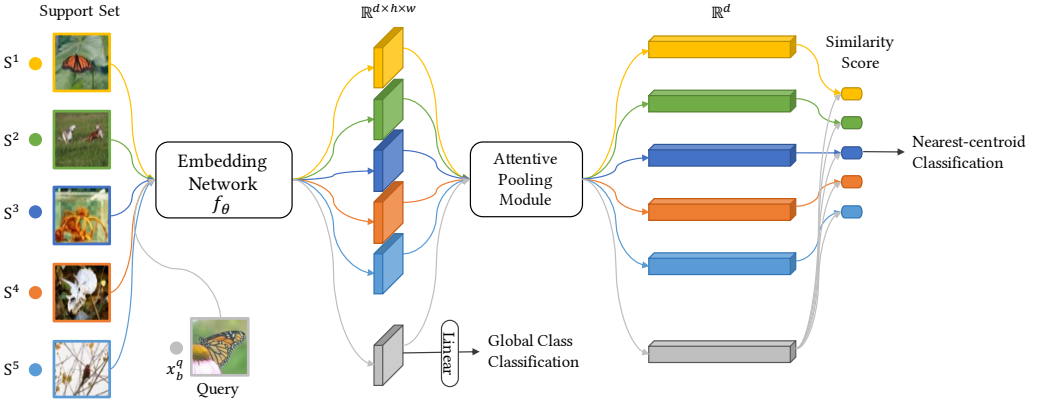


Fig. 2. **Meta-fineting for 5-way 1-shot setting.** Images are first encoded into feature maps by the pre-trained embedding network, and further shrink into image-level embeddings by applying attentive pooling. Losses from both nearest-centroid classification over image-level embeddings and global class classification over local descriptors are added for model optimization.

image-level embedding in \mathbb{R}^d by applying attentive pooling, which takes the form

$$\mathbf{f}_x^{att} = AttPool(f_\theta(\mathbf{x})) = \sum_{j=1}^r \alpha_x^j f_\theta^j(\mathbf{x}), \quad s.t. \alpha_x^j = \frac{A_x^j}{\sum_i A_x^i} \quad (8)$$

where α_x^j denotes the importance coefficient of the j -th local descriptor in the feature map. The attention map A_x , as illustrated in Figure 3, is estimated by a simple regressor f_ϕ^{att} with $A_x^i = \sigma(f_\phi^{att}([\mathbf{f}_x^{GAP}, \mathbf{f}_\theta^i(\mathbf{x})]))$, where $[\cdot]$ is the concatenation operator and σ is the well-known logistic function. Finally, we compute the few-shot classification loss on task \mathcal{T}_i as follows

$$L^{meta} = \frac{-1}{|\mathcal{D}_i^{test}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}_i^{test}} \log(p_x(\hat{y} = y | \mathcal{D}_i^{train})),$$

$$s.t. p_x(\hat{y} = y | \mathcal{D}_i^{train}) = \frac{e^{\langle \mathbf{f}_x^{att}, \mathbf{c}_y / \|\mathbf{c}_y\|_2 \rangle}}{\sum_{j=1}^N e^{\langle \mathbf{f}_x^{att}, \mathbf{c}_j / \|\mathbf{c}_j\|_2 \rangle}} \quad (9)$$

where $p_x(\hat{y} = y | \mathcal{D}_i^{train})$ is the probability of query image \mathbf{x} being classified right. Learnable parameters and the regularization term are omitted in (9) for brevity.

Intuitively, the attentive pooling explains what the learner is looking for as evidence for decision making for few-shot classification. By strengthening the activations of some local descriptors while suppressing the activations of the others, it intends to select more informative local descriptors to build better embeddings for classification. Nevertheless, one challenge we experienced is that the generated attention map A_x tends to be quite unevenly distributed, with one or several local descriptors getting much higher attention scores and in turn overwhelming the whole feature map, leading to information loss during pooling. This is because the attention regressor f_ϕ^{att} works in an unsupervised manner and is implicitly optimized during the overall learning procedure. To avoid this issue, we impose an entropy regularization on the normalized attention map, which takes the

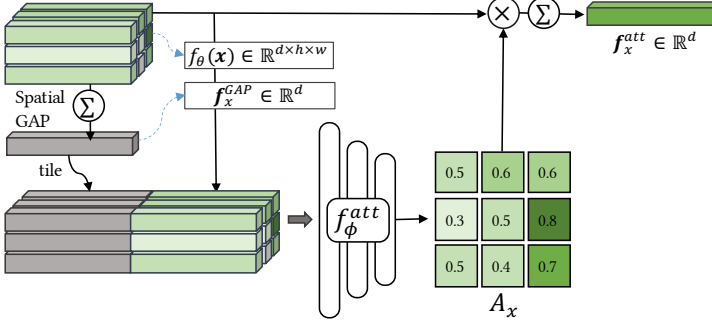


Fig. 3. **Attentive pooling module.** Tensor $f_\theta(x)$ represents the feature map of input image x , and f_x^{GAP} denotes its visual representation after applying GAP. The feature map is seen as $h \times w$ local descriptors in \mathbb{R}^d (here $h = 3, w = 3$) with each representing a local region of the input image. We concatenate each local descriptor and f_x^{GAP} together before feeding the new generated vectors into regressor f_ϕ^{att} to estimate region importances. Local descriptors are scaled accordingly and pooled into an image-level embedding f_x^{att} in \mathbb{R}^d before used for few-shot classification.

form

$$L^{entropy} = \frac{1}{|\mathcal{D}_i^{test}|} \sum_{x \in \mathcal{D}_i^{test}} \sum_{j=1}^r \alpha_x^j \log \alpha_x^j. \quad (10)$$

The regularization guarantees a smoother attention map.

It is also worthy to mention that we simultaneously perform global classification over local descriptors alongside few-shot classification in meta-finetuning. It follows that

$$L^{ce} = \frac{1}{|\mathcal{D}_i^{test}|} \sum_{(x, y_b) \in \mathcal{D}_i^{test}} \sum_{j=1}^r \mathcal{L}^{ce}(\mathbf{W}^T f_\theta^j(x) + \mathbf{b}, y_b), \quad (11)$$

where parameters (\mathbf{W}, \mathbf{b}) are initialized from the pre-trained global classifier and fixed. This dense classification encourages local descriptors to maintain their semantics when fine-tuned for few-shot classification, playing a role of knowledge distillation. We regularize the global classification with label smoothing too. For input image (x, y_b) , the soft label of a local descriptor $f_\theta^j(x)$ is given by setting the smoothing parameter $\epsilon = 1 - A_x^j$. The combination of the aforementioned meta-training loss, entropy regularization and the global classification loss leads to the overall learning objective of meta-finetuning, which takes the form

$$\min_{\Omega} E_{\mathcal{T}_i \in \mathcal{D}_{meta}^{train}} (L^{meta} + \beta L^{entropy} + \gamma L^{ce}) + \mathcal{R}(\Omega), \quad (12)$$

where $\Omega = (\theta, \phi)$ are learnable parameters of both the pre-trained embedding network f_θ and the randomly initialized attention regressor f_ϕ^{att} . (β, γ) are weights to balance the effects of different losses. As we shall see, the meta-finetuning can quickly adapt a pre-trained model to few-shot setting, much faster and better than meta-training from scratch.

4 EXPERIMENTS

In this section, we evaluate the benefits of different components of our proposed DCAP and compare our results with prior arts on two benchmark datasets, *i.e.* *mini*Imagenet [65] and *tiered*Imagenet [45].

4.1 Datasets

miniImagenet. The *miniImagenet* dataset is a derivative of the larger ILSVRC-12 dataset [47]. It consists of 60,000 color images of size 84×84 that are divided into 100 classes with 600 images each. We follow [44, 54] to split the dataset into 64 base classes for meta-training, 16 novel classes for meta-validation, and 20 novel classes for meta-testing, respectively.

tieredImagenet. The *tieredImagenet* dataset is another derivative of the ILSVRC-12 dataset. It contains more than 700,000 images, divided into 608 classes with the average number of images in each class being more than 1200. Particularly, it has a hierarchical structure with all its 608 classes derived from 34 high-level nodes in ILSVRC-12. The 34 top categories are divided into 20 meta-training (351 classes), 6 meta-validation (97 classes) and 8 meta-testing (160 classes) categories. This high-level split provides a more challenging and realistic few-shot setting where the meta-training set is distinctive enough from the meta-testing set semantically.

4.2 Architecture

Embedding Network. We use two backbones as the architecture of our embedding network f_θ for experiments – Resnet12 and Conv4. Resnet12 [19] contains 4 residual blocks of depth 3 with 3×3 kernels. Convolutional layer depth starts with 64 filters and is doubled every next block. A 2×2 max pooling is applied at the end of each block for down-sampling. Likewise, the Conv4 refers to a family of networks that contains 4 convolutional blocks, where each block contains a convolutional layer with 3×3 kernels, a batch normalization layer, a relu activation layer and a 2×2 max pooling layer. Conv4 networks differs from each other in the number of filters in each convolutional block. For example, the number of filters in Conv4-64 is (64, 64, 64, 64). Tab 1 lists 4 frequently used Conv4 variants.

Table 1. 4 frequently used Conv4 variants in the literature.

name	#filters
Conv4-32	(32, 32, 32, 32)
Conv4-64	(64, 64, 64, 64)
Conv4-128	(64, 64, 128, 128)
Conv4-256	(64, 96, 128, 256)

Attention Regressor. The attention regressor f_ϕ^{att} is a plain convolutional network of depth 2 with 8 hidden units in our implementation. The first convolutional layer reduces the embedding space dimension from $d \times 2$ to 8, and the second convolutional layer maps embeddings in the latent space \mathbb{R}^8 into attention scores. Each attention score is a scalar that represents the importance of the corresponding local descriptor. While more sophisticated structures are applicable, it is beyond the scope of this paper.

Global Classifier. The global classifier used in dense classification pre-training is a single fully connected layer with no hidden units. This linear layer maps local descriptors in \mathbb{R}^d into logits in \mathbb{R}^{64} (for *miniImagenet*) or \mathbb{R}^{351} (for *tieredImagenet*). A softmax transformation then turns the logits into probability distributions. In meta-finetuning, the auxiliary global classification adopts a global classifier of the same structure.

4.3 Implementation Details

Pre-training. We train a 64-way global classifier on the meta-training split of *miniImagenet* and a 351-way global classifier on the meta-training split of *tieredImagenet* using standard cross-entropy

loss. In this stage, we follow the conventional deep learning pipeline to horizontally divide each meta-training class into two parts, one for model training and the other for evaluation. The sample rate is set to 0.1. That is, for a meta-training class that has 1000 images in total, we randomly sample 100 images out of the pool for evaluation and use the other 900 images for model training. The learning rate used in this stage is initialized as 0.1. On *miniImagenet*, we train for 100 epochs with a batch size of 256. The learning rate decays once at 80. On *tieredImagenet*, we train for 120 epochs with a batch size of 512, and accordingly the learning rate decays at 50 and 90. We adopt different data augmentations when dealing with the two datasets. Particularly, on *miniImagenet*, we adopt the random horizontal flip data augmentation only. On *tieredImagenet*, both random resized crop and horizontal flip data augmentations are used for model optimization.

Meta-finetuning. The pre-trained embedding network is fine-tuned at a learning rate of 0.001 on the meta-training set that contains 20k/50k tasks randomly sampled from the meta-training split of *miniImagenet*/*tieredImagenet* for few-shot classification. During meta-finetuning, the attention regressor is randomly initialized and optimized with an initial learning rate of 0.01. We follow prior art [21, 28, 74] to consider both 5-way 1-shot and 5-way 5-shot few-shot settings. In 5-way 1-shot setting, a few-shot task has in total 80 images, where each class contains one support image and 15 query images. In 5-way 5-shot setting, the number of query images per class decreases to 10 so as to fit in well with the GPU memory. We consider a mini-batch that consists of 4 few-shot tasks and adopt the same data augmentation (*i.e.* random horizontal flip) when dealing with both datasets. The learning rate decays once at 16k and 40k on *miniImagenet* and *tieredImagenet* respectively.

Evaluation Protocols. To be consistent with previous work [17, 53, 62, 77], we construct a meta-evaluation/meta-testing set by randomly sampling 1000 few-shot tasks from the meta-validation/meta-testing split of each dataset for model validation/testing. Mean accuracy along with the 95% confidence interval over the 1000 tasks is reported for model comparison. Besides, we follow [7] to apply *consistent sampling* when preparing the meta-evaluation/meta-testing set. Consistent sampling ensures that the same set of tasks are sampled for model evaluation/testing in each epoch following a deterministic order.

All our experiments are implemented in PyTorch⁴ on an Ubuntu server with 4 NVIDIA TITAN Xp GPU cards. In both pre-training and meta-finetuning, we adopt SGD with Nesterov momentum of 0.9 and weight decay of 0.0005 for model optimization. The learning rate decay is set to 0.1 in two stages. Unless otherwise specified, we set $\beta = 0.1$ and $\gamma = 0.5$ in (12) in all experiments.

4.4 Results with ResNet12 Backbone

Tab 2 compares our method with other few-shot algorithms when ResNet12 is used as the backbone network. As it can be seen, our proposed DCAP can consistently perform better than prior art on *miniImagenet* and *tieredImagenet* under both 5-way 1-shot and 5-way 5-shot settings. Specifically, our method is around 3%/6.4%, 4%/5.4%, 2.7%/2%, 1.4%/1.5%, 3.5%/2.6%, 2%/1.7%, 2.6%/2.3%, 3.2%/1.3%, 2.6%/2.1%, 1%/0.9% better than Fine-tuning [8], MTL [57], DC [34], CAN [21], FEAT [74], Meta-Baseline [7], MetaOptNet [28], RFS [62], DSN [53], Meta-UAFS [77] under the 1-shot/5-shot setting on *miniImagenet*, respectively. On *tieredImagenet*, our method is around 1.7%/2%, 0.2%/1%, 1.5%/2%, 4.1%/3.8%, 0.4%/0.9%, 3.9%/2.6%, 1%/1% better than Fine-tuning [8], CAN [21], Meta-Baseline [7], MetaOptNet [28], RFS [62], DSN [53], Meta-UAFS [77], respectively. Notably, we observe that Prototypical Networks provide a strong baseline. It not only outperforms SNAIL [40] and DN4 [31] by a large margin, but also has comparable results against TADAM [42] and even TPN [37], a transductive method in which all query images arrive simultaneously during inference. In principle, transductive methods are able to achieve higher performances than inductive methods

⁴PyTorch: <https://pytorch.org>

Table 2. **Comparison to prior work on *minilImageNet* and *tieredImageNet* (ResNet12).** Mean accuracies (%) with 95% confidence intervals on *minilImageNet* and *tieredImageNet* meta-testing set. † models use transductive inference. * Results from [53]. ‡ refers to our reimplementations.

model	backbone	<i>minilImageNet</i> 5-way		<i>tieredImageNet</i> 5-way	
		1-shot	5-shot	1-shot	5-shot
Prototypical Networks* [54]	ResNet-12	59.25 ± 0.64	75.60 ± 0.48	61.74 ± 0.77	80.00 ± 0.55
Prototypical Networks‡ [54]	ResNet-12	59.25 ± 0.70	76.88 ± 0.59	62.15 ± 0.71	81.72 ± 0.53
SNAIL [40]	ResNet-12	55.71 ± 0.99	68.88 ± 0.92	-	-
TADAM [42]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30	-	-
DN4 [31]	ResNet-12	54.37 ± 0.36	74.44 ± 0.29	-	-
TPN† [37]	ResNet-12	59.46 ± -	75.65 ± -	-	-
Fine-tuning [8]	ResNet-12	56.67 ± 0.62	74.70 ± 0.51	64.45 ± 0.70	83.59 ± 0.51
Fine-tuning† [8]	ResNet-12	62.35 ± 0.66	74.53 ± 0.54	68.41 ± 0.73	83.41 ± 0.52
MTL [57]	ResNet-12	61.20 ± 1.80	75.50 ± 0.80	-	-
DC [34]	ResNet-12	62.53 ± 0.19	78.95 ± 0.13	-	-
CAN [21]	ResNet-12	63.85 ± 0.48	79.44 ± 0.34	69.89 ± 0.51	84.23 ± 0.37
FEAT* [74]	ResNet-12	61.72 ± 0.11	78.32 ± 0.76	-	-
Meta-Baseline [7]	ResNet-12	63.17 ± 0.23	79.26 ± 0.17	68.62 ± 0.27	83.29 ± 0.18
MetaOptNet [28]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
RFS [62]	ResNet-12	62.02 ± 0.63	79.64 ± 0.44	69.74 ± 0.72	84.41 ± 0.55
DSN [53]	ResNet-12	62.64 ± 0.66	78.83 ± 0.45	66.22 ± 0.75	82.79 ± 0.48
Meta-UAFS [77]	ResNet-12	64.22 ± 0.67	79.99 ± 0.49	69.13 ± 0.84	84.33 ± 0.59
DCAP (ours)	ResNet-12	65.20 ± 0.67	80.93 ± 0.53	70.15 ± 0.74	85.33 ± 0.55

because the transductive methods can leverage information in other query images when dealing with the current query image while the inductive methods can not.

Note that, RFS and DSN reported in Tab 2 are not the best models as reported in the original papers. In [62], RFS exploited Born-again [12] strategy to apply knowledge distillation sequentially to revise the embedding network and its associated base learner multiple times. The strategy is not adopted by other works, including ours of course. For fair comparison, we report the results of its variant without Born-again strategy in the table. DSN [53] proposed to refine the prototype of each class by taking advantage of unlabeled data when addressing semi-supervised few-shot classification problem and achieved outstanding performance. In this work, we consider the standard few-shot classification setting but leave the semi-supervised setting untouched. Therefore, we also choose to report results of its variant that studies the same standard few-shot classification for fair comparison.

4.5 Results with Conv4 Backbone

To give an overview of the literature, we additionally take a Conv4 as the backbone network, evaluate our method on *minilImageNet* and *tieredImageNet* under both *5-way 1-shot* and *5-way 5-shot* settings, and compare our method with other works that adopt the same or similar backbone in Tab 3. Particularly, we follow [17, 53, 54, 58] to adopt the widely used Conv4-64 as described in Tab 1 for experiments. The Conv4-64 maps an input image of size 84×84 into a feature map of size $64 \times 5 \times 5$. As can be seen, our method can achieve competitive or better performance when compared to most of the counterparts. For example, our method is around 10%/13%, 2.3%/1.5%, 4%/2.2%, 3.3%/2.2% better than MAML [11], Relation Networks [58], Fine-tuning [8], GNN [50]

Table 3. **Comparison to prior work on *miniImageNet* and *tieredImageNet* (Conv4).** Mean accuracies (%) with 95% confidence intervals on *miniImageNet* and *tieredImageNet* meta-testing set. a-b-c-d denotes a Conv4 network with a, b, c, and d filters in each convolutional block. [†] models use transductive inference. [‡] refers to our reimplementation.

model	backbone	<i>miniImageNet</i> 5-way		<i>tieredImageNet</i> 5-way	
		1-shot	5-shot	1-shot	5-shot
MAML [11]	32-32-32-32	48.70 ± 1.84	63.11 ± 0.92	51.67 ± 1.81	70.30 ± 1.75
Meta-SGD [33]	32-32-32-32	50.47 ± 1.87	64.03 ± 0.94	-	-
Meta-Learner [44]	32-32-32-32	43.44 ± 0.77	60.60 ± 0.71	-	-
Reptile [41]	32-32-32-32	49.97 ± 0.32	65.99 ± 0.58	-	-
TAML [23]	32-32-32-32	49.33 ± 1.8	66.05 ± 0.85	-	-
Matching Networks [65]	64-64-64-64	43.56 ± 0.84	55.31 ± 0.73	-	-
Prototypical Networks [54]	64-64-64-64	46.14 ± 0.77	65.77 ± 0.70	48.58 ± 0.87	69.57 ± 0.75
Prototypical Networks [‡] [54]	64-64-64-64	49.76 ± 0.60	67.63 ± 0.58	51.32 ± 0.68	70.33 ± 0.59
Relation Networks [58]	64-64-64-64	51.38 ± 0.82	67.07 ± 0.69	54.48 ± 0.93	71.31 ± 0.78
DN4 [31]	64-64-64-64	51.24 ± 0.74	71.02 ± 0.64	-	-
MetaOptNet [28]	64-64-64-64	52.87 ± 0.57	68.76 ± 0.48	54.71 ± 0.67	71.79 ± 0.59
PARN [68]	64-64-64-64	55.22 ± 0.84	71.55 ± 0.66	-	-
Fine-tuning [8]	64-64-64-64	49.43 ± 0.62	66.42 ± 0.53	57.45 ± 0.68	73.96 ± 0.56
Fine-tuning [†] [8]	64-64-64-64	50.46 ± 0.62	66.68 ± 0.52	58.05 ± 0.68	74.24 ± 0.56
TPN [†] [37]	64-64-64-64	53.75 ± 0.86	69.43 ± 0.67	57.53 ± 0.96	72.85 ± 0.74
MRN [†] [17]	64-64-64-64	57.83 ± 0.69	71.13 ± 0.50	62.65 ± 0.84	74.20 ± 0.64
DSN [53]	64-64-64-64	51.78 ± 0.96	68.99 ± 0.69	-	-
GNN [50]	64-96-128-256	50.33 ± 0.36	66.41 ± 0.63	-	-
DCAP (ours)	64-64-64-64	53.68 ± 0.63	68.62 ± 0.60	55.63 ± 0.73	70.78 ± 0.60
	64-64-128-128	55.25 ± 0.61	70.46 ± 0.60	58.75 ± 0.73	72.81 ± 0.61

under *1-shot/5-shot* setting on *miniImageNet*, respectively. We notice that our method is a little bit inferior to prior art DN4 [31], MetaOptNet [28] and DSN [53] in *5-shot* setting. It is also observed that our method underperforms transductive methods, like TPN [37] and Fine-tuning [8], especially on *tieredImageNet*, which is inconsistent with our observation in §4.4. We argue the reason of the inferiority is two-fold.

- First, dense classification pre-training the light-weight Conv4-64 backbone is more difficult because of its limited representation ability. In the pre-training stage, by training the Conv4-64 backbone on the meta-training split of *miniImageNet/tieredImageNet* for dense classification, we can merely get a top-1 evaluation accuracy of 33.08%/12.21%. In contrast, pre-training the ResNet12 backbone in this stage gives a top-1 evaluation accuracy of 64.57%/56.88%. The poorly pre-trained backbone explains why the superiority of our method is not preserved when switching from ResNet12 to Conv4-64. Even so, we are surprised to observe that the pre-trained backbone can achieve results that are not bad, or even better than previous work in *1-shot* setting.
- Second, a 64-dimensional feature embedding is much less representative than a 1600-dimensional one. A common practice adopted by prior arts when working with Conv4-64 is to flatten feature maps in $\mathcal{R}^{64 \times 5 \times 5}$ to construct 1600-dimensional feature embeddings and use the high-dimensional embeddings in the final classifier for few-shot classification. The resulting embeddings keep all information that is originally available in the feature maps, including

spatial information and semantic information, leading to no information loss. In this work, we provide an alternative way to build embeddings by applying attentive pooling over the feature maps, which attentively pools feature maps into 64-dimensional embeddings. While attentive pooling can select to emphasize important local descriptors, the pooling, like GAP, would inevitably result in information loss. As a result, using the 1600-dimensional embeddings for few-shot classification unquestionably gains superior results in some cases.

To validate our assumptions, we use a larger Conv4-128 backbone for experiments and find that an average of 2% improvement can be obtained on all settings.

4.6 Cross-Domain Few-Shot Classification

We further justify the effectiveness of our proposed method under a more challenging cross-domain few-shot classification setting, where the learner is trained and tested on few-shot tasks collected from different domains. To be specific, we follow [1, 6, 67] to consider the *miniImagenet* \rightarrow *CUB* cross-domain scenario, where the learner is trained on tasks sampled from the *miniImagenet* dataset but evaluated on tasks sampled from the CUB-200-2011 dataset [5]. The CUB-200-2011 dataset (also referred as CUB) contains 11,788 images from 200 bird classes and is originally constructed to serve fine-grained bird classification. In few-shot setting, the CUB dataset is divided into 100 classes for meta-training, 50 classes for meta-validation and 50 classes for meta-testing. We randomly sample 1,000 few-shot tasks from its meta-testing split for cross-domain evaluation. Results are reported in Tab 4. As can be seen, few-shot classification approaches normally suffer significant performance drop in the presence of domain shift between source and target datasets. Nonetheless, our method keeps outperforming many other participants in both 5-way 1-shot and 5-way 5-shot cross-domain few-shot settings.

4.7 Ablation Study

To better understand our method, we analyze DCAP and the following ablated variants on the *miniImagenet* dataset with ResNet12 backbone.

- **DC-GAP** refers to the model that is trained to solve dense classification in pre-training. In meta-finetuning, it pools feature maps into embeddings by applying GAP.
- **GAP-AttPool** refers to the model that is trained to solve standard classification in pre-training where feature maps are pooled into embeddings by applying GAP. In meta-finetuning, it pools feature maps into embeddings by applying attentive pooling.
- **GAP-GAP**, similar to DC-GAP and GAP-AttPool, is trained to solve standard classification in pre-training and applies GAP over feature maps in meta-finetuning. It shares the same pipeline with [7].
- **Zero-GAP** refers to the model that is meta-trained from scratch where feature maps are pooled into embeddings by applying GAP.
- **Zero-AttPool** similarly denotes the model that is meta-trained from scratch where feature maps are pooled into embeddings by applying attentive pooling.

We follow the naming scheme to give our DCAP the alias **DC-AttPool**. The results of different models under *1-shot* setting are reported in Tab 5 for straightforward comparisons.

Is pre-training stage necessary? As shown in Tab 5, meta-learning on top of a pre-trained model can bring 4.3% \sim 8.5% improvements over models meta-trained from scratch, which demonstrates the importance of pre-training in few-shot classification. Concretely, the GAP-GAP/DC-GAP model is about 4.3%/4.9% better than the Zero-GAP model, and the DC-GAP/DC-AttPool model is around 6.9%/8.5% better than the Zero-AttPool model. We argue the pre-training provides a model with strong representation ability, whereas the beneficial ability is not fully accessible by

Table 4. **Performance comparison in the cross-domain setting: *minilmagenet* → CUB.** Experiments are performed with both two types of backbones. Symbols and organization match Tab 3. "+FT" indicates the model is fine-tuned towards the target task in evaluation.

model	backbone	<i>miniImageNet</i> → CUB	
		5-way 1-shot	5-way 5-shot
MAML [6, 11]	ResNet-18	-	51.34 ± 0.72
Prototypical Networks [6, 54]	ResNet-18	-	62.02 ± 0.70
Baseline [6]	ResNet-18	-	65.57 ± 0.70
Baseline++ [6]	ResNet-18	-	62.04 ± 0.76
Matching Networks+FT [63, 67]	ResNet-10	36.61 ± 0.53	55.23 ± 0.83
Relation Networks+FT [63, 67]	ResNet-10	44.07 ± 0.77	59.46 ± 0.71
GNN+FT [63, 67]	ResNet-10	47.47 ± 0.75	66.98 ± 0.68
MetaOptNet [28, 38, 67]	ResNet-12	44.79 ± 0.75	64.98 ± 0.68
MAML+L2F [1]	ResNet-12	-	62.12 ± 0.21
MAML+ALFA [1]	ResNet-12	-	61.22 ± 0.22
ALFA+Random Init [1]	ResNet-12	-	60.13 ± 0.23
ALFA+MAML+L2F [1]	ResNet-12	-	63.24 ± 0.22
DCAP (ours)	ResNet-12	49.65 ± 0.70	68.25 ± 0.65
MAML [11, 55]	32-32-32-32	34.01 ± 1.25	48.83 ± 0.62
Bayesian MAML [55, 75]	32-32-32-32	33.52 ± 0.36	51.35 ± 0.16
Matching Networks [55, 65]	64-64-64-64	36.98 ± 0.06	50.72 ± 0.36
Prototypical Networks [54, 55]	64-64-64-64	33.27 ± 1.09	52.16 ± 0.17
Relation Networks [55, 58]	64-64-64-64	37.13 ± 0.20	51.76 ± 1.48
DCAP (ours)	64-64-64-64	39.66 ± 0.54	52.22 ± 0.53

Table 5. **5-way 1-shot accuracy on *minilmagenet*.** In pre-training, "GAP" refers to standard classification upon image-level embeddings generated by applying GAP. "DC" refers to dense classification over local descriptors. In meta-learning, "GAP" and "AttPool" stand for global average pooling and attentive pooling, respectively. We report the results of meta-training from scratch in the first row.

Stage 1: Pre-training			Stage 2: Meta-learning	
w/pre-training	GAP	DC	GAP	AttPool
			59.25 ± 0.70	56.69 ± 0.68
✓	✓		63.58 ± 0.66	63.57 ± 0.66
✓		✓	64.13 ± 0.69	65.20 ± 0.67

straight meta-training on few-shot tasks. Since the meta-training focuses more on transferability, *i.e.* the generalization ability across tasks, a model that starts from a randomly initialized state would struggle to balance the representation ability and the transferability during meta-training, leading to inferior performance. From Tab 5, our another observation is that models with attentive pooling are more sensitive to pre-training. We think this is due to the difficulty of optimizing the attention regressor for the selection of informative local descriptors if the model is meta-trained from scratch.

Is meta-finetuning stage necessary? Chen *et al.* [7] suggested that competitive performance could be obtained on top of a pre-trained model without meta-finetuning. In this work, we have

Table 6. **5-way 1-shot accuracy on *minilmagenet* without meta-finetuning.** Results of models with attentive pooling are not given because the attention regressor is not trained yet.

Stage 1: Pre-training			GAP	AttPool
w/pre-training	GAP	DC		
			26.50 ± 0.46	26.21 ± 0.46
✓	✓		20.14 ± 0.30	-
✓		✓	20.14 ± 0.08	-

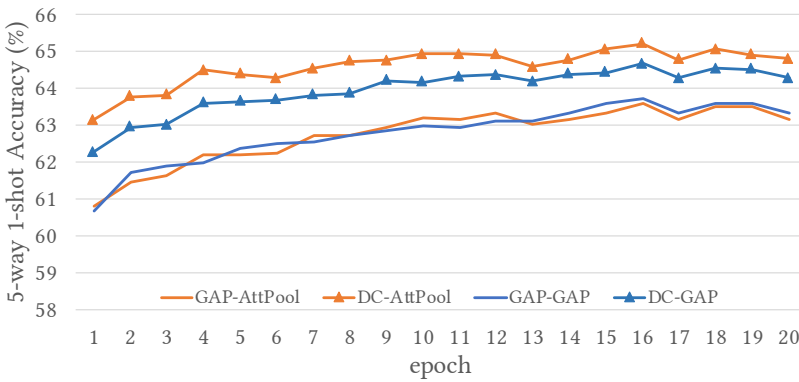


Fig. 4. **5-way 1-shot accuracy on *minilmagenet* with meta-finetuning.** Dense classification pre-trained models are marked with Δ . Models that utilize the same pooling in meta-finetuning are colored the same. Best viewed in color.

different findings. After the pre-training and right before meta-finetuning, we apply GAP over feature maps and use the resulting embeddings to perform few-shot classification as described in §3.2. The mean accuracies over 1000 meta-testing tasks are reported in Tab 6. It is astonishing to observe that the both DC-pretrained and GAP-pretrained models perform just slightly better than random guessing or even worse than random initialized models. On the contrary, Fig 4 shows that the models can achieve $> 60\%$ accuracies after meta-finetuning for one epoch. It indicates that meta-learning/meta-finetuning is at the core of the success of few-shot classification.

Influence of dense classification pre-training. Tab 5 and Fig 4 show that meta-finetuning a DC-pretrained model generally achieves higher accuracies. To identify the reason, we investigate the effect of dense classification and find that the dense classification pre-training is able to produce smoother feature maps. The smoothness is evidenced by (i) local descriptors in the feature map generated by the DC-pretrained model have L_2 -norm values with a smaller deviation, and (ii), as shown in Fig 1 and Fig 5, a local descriptor in the feature map generated by the DC-pretrained model has higher cosine similarities with other local descriptors in its 8-connected neighborhood, demonstrating some kind of semantic consistency. We hypothesize it is exact the semantic inconsistency between local descriptors in GAP-pretrained feature maps that makes it harder to meta-finetune for fast adaptation.

Influence of attentive pooling. Attentive pooling aims to select informative local descriptors for constructing better embeddings for few-shot classification. As can be seen in Tab 5, applying attentive pooling when meta-finetuning a DC-pretrained model for few-shot classification gives the best 5-way 1-shot accuracy on *minilmagenet*. However, we notice that performance degradation

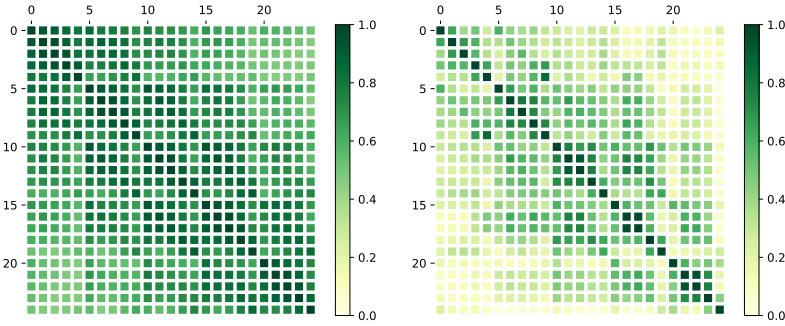


Fig. 5. **Cosine similarities between local descriptors of a test image.** Left: similarities between local descriptors in the feature map generated by DC-pretrained model. Right: similarities between local descriptors in the feature map generated by GAP-pretrained model.

Table 7. **5-way 1-shot accuracy on *minilmagenet* with different loss weights.** We report the accuracies when meta-training from scratch with different loss weights. η and γ balance the effects of meta-learning and global classification.

$\mathcal{L}^{meta}(\eta)$	$\mathcal{L}^{ce}(\gamma)$	GAP	AttPool
1.0	0.5	59.25 \pm 0.70	56.69 \pm 0.68
0.5	1.0	59.21 \pm 0.65	58.42 \pm 0.70
1.0	0.0	56.39 \pm 0.68	56.51 \pm 0.74

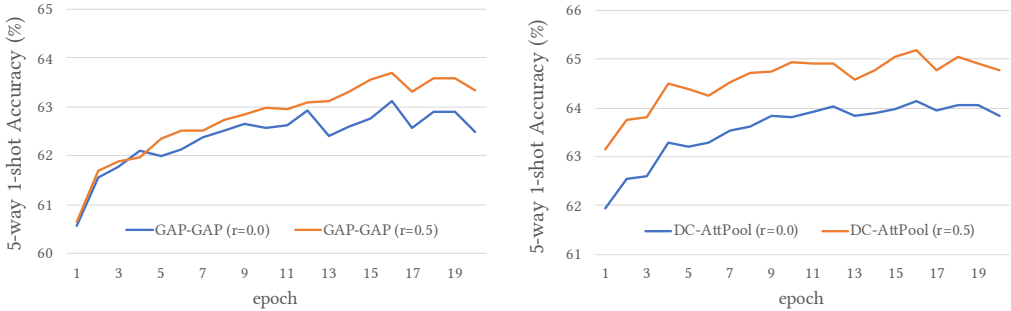


Fig. 6. **5-way 1-shot accuracy on *minilmagenet* with/without global classification.** Left: meta-finetuning the GAP-GAP model with or without global classification. Right: meta-finetuning the DC-AttPool model with or without global classification. γ is the loss weight of global classification in (12).

happens when meta-training from scratch (59.25% vs 56.69%). We conjecture that the reason is two-fold. First, dense classification over local descriptors is much more difficult than standard image classification over image-level representations. As a result, Zero-AttPool struggles harder than Zero-GAP to balance the global classification and the few-shot classification when meta-trained from scratch, resulting in lower accuracy. We confirm this finding by disabling global classification (*i.e.* $\gamma = 0$) in meta-training. As shown in Tab 7, when the global classification is disabled, Zero-AttPool runs neck-and-neck with Zero-GAP. We also find that the performance gap between

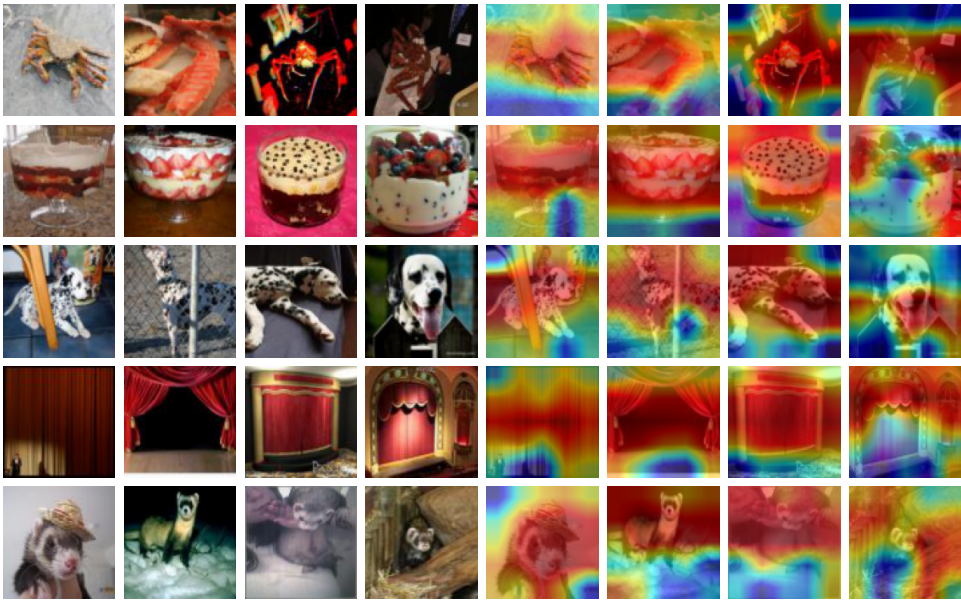


Fig. 7. An example of the attention maps generated by the attention regressor. The regressor can produce attention maps that focus on the target object in image and diminish the impact of background. In each row, we report a failure case in the last column. Warmer color indicates higher score.

Zero-AttPool and Zero-GAP can be reduced if increasing the loss weight γ of global classification. Second, the attention regressor cannot be efficiently optimized in meta-training from scratch. This is because local descriptors in the very beginning are of no semantic information. From the chaos, it is challenging for the regressor to figure out which descriptor is more important. In other word, the regressor is error-prone which does harm to the global classification and the few-shot classification. Only when starting from a pre-trained state, models can benefit from attentive pooling.

Influence of global classification. We perform global classification in parallel with few-shot classification to enhance the representation ability of feature embeddings and encourage local descriptors to maintain their semantics during meta-training. Tab 7 shows that the joint learning is beneficial when meta-training from scratch. To investigate the effect of global classification in our two-stage training, we examine models without global classification and plot accuracies in Fig 6. As shown in the figure, global classification can consistently bring improvements when dealing with both the GAP-GAP model and the DC-AttPool model. Especially, the effect of global classification on our DC-AttPool is more obvious.

4.8 Visualization

Our method highly relies on the attention regressor to correctly estimate the importance of each local descriptor within a feature map in order to select informative ones for few-shot classification. To check whether the regressor works as expected, we investigate the generated attention maps. Unsurprisingly, we find that the regressor can learn to concentrate on the target object in image by assigning higher attention scores to the corresponding descriptors. And conversely, local descriptors that are related to distractors or the background normally get lower attention scores. For visualization, we present an example of the attention maps generated by the regressor in Fig 7. As illustrated, for the three images from class *crab*, the attention maps can capture the spatial

information and highlight the target objects properly. But we have to admit that the attention regressor is not perfect yet. It sometimes makes mistakes in which it fails to make an estimation that is consistent with our awareness. We conjecture that the failure results from illumination, view-point, *etc.*

5 CONCLUSION AND FUTURE WORK

We propose a new method, namely DCAP, to tackle the few-shot classification problem. Instead of designing complex classifiers as in prior art, in this work, we propose to adopt a simple nearest-centroid classifier and approach few-shot classification from the perspective of acquiring more discriminative and robust embeddings. Our DCAP is optimized following a two-stage training paradigm. In the pre-training stage, we propose to train the embedding network to solve dense classification problem, and in meta-finetuning we fine-tune the pre-trained model on a family of few-shot tasks to perform few-shot classification. An attention regressor is further devised in meta-finetuning to reweight local descriptors when summarizing them into image-level embeddings for few-shot classification. We find that dense classification effectively eliminates semantic discrepancy among local descriptors that is observed in previous work. And the attention regressor learns to select informative local descriptors to build better embeddings. By meta-finetuning a dense classification pre-trained model with attentive pooling, we reveal that the simple nearest-centroid classifier is able to yield accuracies that exceed most of the leading methods on two standard benchmark datasets. In the future, we will try to apply our proposed method for other visual tasks, such as few-shot visual dialog [14, 15] and few-shot visual relationship detection [32, 51].

ACKNOWLEDGMENT

This research was supported by the National Key R&D Program of China under Grant 2019YFA0706203, the National Natural Science Foundation of China (NSFC) under grants 61932009 and 61976076, and A*STAR under its AME YIRG Grant (Project No. A20E6c0101). The authors would like to thank Shengen Tang, Huixia Ben, as well as anonymous reviewers for their careful reading of the manuscript and for providing constructive feedback and valuable suggestions.

REFERENCES

- [1] Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. 2020. Meta-Learning with Adaptive Hyperparameters. In *Proceedings of the 33th International Conference on Neural Information Processing Systems*, Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).
- [2] Nihar Bendre, Hugo Terashima-Marín, and Peyman Najafirad. 2020. Learning from Few Samples: A Survey. *CoRR abs/2007.15484* (2020).
- [3] Yi Bin, Yang Yang, Jie Zhou, Zi Huang, and Heng Tao Shen. 2017. Adaptively attending to visual attributes and linguistic knowledge for captioning. In *Proceedings of the 25th ACM international conference on multimedia*. 1345–1353.
- [4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).
- [5] Wah Catherine, Branson Steve, Welinder Peter, Perona Pietro, and Belongie Serge. 2011. *The Caltech-UCSD Birds-200-2011 Dataset*. Technical Report.
- [6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A Closer Look at Few-shot Classification. In *7th International Conference on Learning Representations, 2019*. OpenReview.net.
- [7] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. 2020. A new meta-baseline for few-shot learning. *arXiv:2003.04390* (2020).
- [8] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. 2019. A Baseline for Few-Shot Image Classification. In *International Conference on Learning Representations*.
- [9] Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*. 1422–1430.
- [10] Carl Doersch, Ankush Gupta, and Andrew Zisserman. 2020. CrossTransformers: spatially-aware few-shot transfer. *arXiv preprint arXiv:2007.11498* (2020).

- [11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 1126–1135.
- [12] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-Again Neural Networks. In *Proceedings of the 35th International Conference on Machine Learning*, Jennifer G. Dy and Andreas Krause (Eds.), Vol. 80. 1602–1611.
- [13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning*.
- [14] Dan Guo, Hui Wang, and Meng Wang. 2021. Context-Aware Graph Inference with Knowledge Distillation for Visual Dialog. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).
- [15] Dan Guo, Hui Wang, Hanwang Zhang, Zheng-Jun Zha, and Meng Wang. 2020. Iterative context-aware graph inference for visual dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10055–10064.
- [16] Bharath Hariharan and Ross B. Girshick. 2017. Low-Shot Visual Recognition by Shrinking and Hallucinating Features. In *International Conference on Computer Vision*. 3037–3046.
- [17] Jun He, Richang Hong, Xueliang Liu, Mingliang Xu, Zheng-Jun Zha, and Meng Wang. 2020. Memory-Augmented Relation Network for Few-Shot Learning. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1236–1244.
- [18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
- [20] Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. 2020. Meta-Learning in Neural Networks: A Survey. *CoRR* abs/2004.05439 (2020).
- [21] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. 2019. Cross attention network for few-shot classification. *Advances in Neural Information Processing Systems* 32 (2019), 4003–4014.
- [22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.
- [23] Muhammad Abdullah Jamal and Guo-Jun Qi. 2019. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11719–11727.
- [24] Zihang Jiang, Bingyi Kang, Kuangqi Zhou, and Jiashi Feng. 2020. Few-shot Classification via Adaptive Attention. *arXiv:2008.02465* (2020).
- [25] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. 2019. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11–20.
- [26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.
- [27] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Learning representations for automatic colorization. In *European conference on computer vision*. Springer, 577–593.
- [28] Kwonjoon Lee, Subhansu Maji, Avinash Ravichandran, and Stefano Soatto. 2019. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10657–10665.
- [29] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. 2020. Boosting few-shot learning with adaptive margin loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12576–12584.
- [30] Suichan Li, Dapeng Chen, Bin Liu, Nenghai Yu, and Rui Zhao. 2019. Memory-based neighbourhood embedding for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6102–6111.
- [31] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. 2019. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7260–7268.
- [32] Yicong Li, Xun Yang, Xindi Shang, and Tat-Seng Chua. 2021. Interventional video relation detection. In *ACM International Conference on Multimedia*. 4091–4099.
- [33] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. 2017. Meta-SGD: Learning to Learn Quickly for Few Shot Learning. *CoRR* abs/1707.09835 (2017).
- [34] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. 2019. Dense classification and implanting for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9258–9267.
- [35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.
- [36] Xueliang Liu, Xun Yang, Meng Wang, and Richang Hong. 2020. Deep neighborhood component analysis for visual similarity modeling. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11, 3 (2020), 1–15.

- [37] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. 2019. Learning to Propagate Labels: Transductive Propagation Network for Few-Shot Learning. In *International Conference on Learning Representations*.
- [38] Puneet Mangla, Mayank Singh, Abhishek Sinha, Nupur Kumari, Vineeth N. Balasubramanian, and Balaji Krishnamurthy. 2020. Charting the Right Manifold: Manifold Mixup for Few-shot Learning. In *IEEE Winter Conference on Applications of Computer Vision, 2020*. IEEE, 2207–2216.
- [39] Lei Meng, Long Chen, Xun Yang, Dacheng Tao, Hanwang Zhang, Chunyan Miao, and Tat-Seng Chua. 2019. Learning using privileged information for food recognition. In *Proceedings of the 27th ACM International Conference on Multimedia*. 557–565.
- [40] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2018. A Simple Neural Attentive Meta-Learner. In *6th International Conference on Learning Representations, ICLR 2018*.
- [41] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On First-Order Meta-Learning Algorithms. *CoRR* abs/1803.02999 (2018).
- [42] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*. 721–731.
- [43] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. 2019. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv:1909.09157* (2019).
- [44] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a Model for Few-Shot Learning. In *Proceedings of International Conference on Learning Representations*.
- [45] Mengye Ren, Eleni Triantafyllou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. 2018. Meta-Learning for Semi-Supervised Few-Shot Classification. In *International Conference on Learning Representations*.
- [46] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. 2020. Embedding propagation: Smoother manifold for few-shot classification. In *European Conference on Computer Vision*. Springer, 121–138.
- [47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115, 3 (2015), 211–252.
- [48] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. 2019. Meta-Learning with Latent Embedding Optimization. In *International Conference on Learning Representations*.
- [49] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. 2016. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of the 33rd International Conference on Machine Learning*, Vol. 48. 1842–1850.
- [50] Victor Garcia Satorras and Joan Bruna Estrach. 2018. Few-Shot Learning with Graph Neural Networks. In *International Conference on Learning Representations*.
- [51] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. 2019. Annotating objects and relations in user-generated videos. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. 279–287.
- [52] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. 2019. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7912–7921.
- [53] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. 2020. Adaptive Subspaces for Few-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4136–4145.
- [54] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems* 30 (2017), 4077–4087.
- [55] Jake Snell and Richard S. Zemel. 2021. Bayesian Few-Shot Classification with One-vs-Each Pólya-Gamma Augmented Gaussian Processes. In *International Conference on Learning Representations, 2021*.
- [56] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. 2020. When does self-supervision improve few-shot learning?. In *European Conference on Computer Vision*. Springer, 645–666.
- [57] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. 2019. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 403–412.
- [58] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1199–1208.
- [59] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2818–2826.
- [60] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 6105–6114.

- [61] Yi Tan, Yanbin Hao, Xiangnan He, Yinwei Wei, and Xun Yang. 2021. Selective Dependency Aggregation for Action Classification. In *Proceedings of the 29th ACM International Conference on Multimedia*. 592–601.
- [62] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. 2020. Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need? *arXiv:2003.11539* (2020).
- [63] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. 2020. Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation. In *International Conference on Learning Representations, 2020*.
- [64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in neural information processing systems*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), 5998–6008.
- [65] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 3637–3645.
- [66] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. 2018. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7278–7286.
- [67] Davis Wertheimer, Luming Tang, and Bharath Hariharan. 2021. Few-Shot Classification With Feature Map Reconstruction Networks. In *IEEE Conference on Computer Vision and Pattern Recognition, 2021*. 8012–8021.
- [68] Ziyang Wu, Yuwei Li, Lihua Guo, and Kui Jia. 2019. Parn: Position-aware relation networks for few-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 6659–6667.
- [69] Hui Xu, Jiaying Wang, Hao Li, Deqiang Ouyang, and Jie Shao. 2021. Unsupervised meta-learning for few-shot learning. *Pattern Recognition* 116 (2021), 107951.
- [70] Hui Xu, Chong Zhang, Jiaying Wang, Deqiang Ouyang, Yu Zheng, and Jie Shao. 2021. Exploring parameter space with structured noise for meta-reinforcement learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 3153–3159.
- [71] Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua. 2021. Deconfounded Video Moment Retrieval with Causal Intervention. In *ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [72] Xun Yang, Shanshan Wang, Jian Dong, Jianfeng Dong, Meng Wang, and Tat-Seng Chua. 2022. Video Moment Retrieval with Cross-Modal Neural Architecture Search. *IEEE Transactions on Image Processing* 31 (2022), 1204–1216.
- [73] Xun Yang, Peicheng Zhou, and Meng Wang. 2018. Person reidentification via structural deep metric learning. *IEEE transactions on neural networks and learning systems* 30, 10 (2018), 2987–2998.
- [74] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. 2020. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8808–8817.
- [75] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. 2018. Bayesian Model-Agnostic Meta-Learning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, 2018*. 7343–7353.
- [76] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. 2021. Self-Regulation for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6953–6963.
- [77] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Zhibo Chen, and Shih-Fu Chang. 2020. Uncertainty-Aware Few-Shot Image Classification. *arXiv:2010.04525* (2020).
- [78] Yuanen Zhou, Meng Wang, Daqing Liu, Zhenzhen Hu, and Hanwang Zhang. 2020. More grounded image captioning by distilling image-text matching model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4777–4786.