Adaptive Subspaces for Few-Shot Learning

Christian Simon[†,§] Piotr Koniusz[†,§] Richard Nock[†,‡,§] Mehrtash Harandi[♣,§]
[†]The Australian National University, [♣]Monash University,
[‡]The University of Sydney, [§] Data61-CSIRO
first.last@{anu.edu.au,monash.edu,data61.csiro.au}

Abstract

*Object recognition requires a generalization capability to avoid overfitting, especially when the samples are extremely few. Generalization from limited samples, usually studied under the umbrella of meta-learning, equips learning techniques with the ability to adapt quickly in dynamical environments and proves to be an essential aspect of lifelong learning. In this paper, we provide a framework for few-shot learning by introducing dynamic classifiers that are constructed from few samples. A subspace method is exploited as the central block of a dynamic classifier. We will empirically show that such modelling leads to robustness against perturbations (*e.g.*, outliers) and yields competitive results on the task of supervised and semi-supervised few-shot classification. We also develop a discriminative form which can boost the accuracy even further. Our code is available at* https://github.com/chrysts/dsn_fewshot.

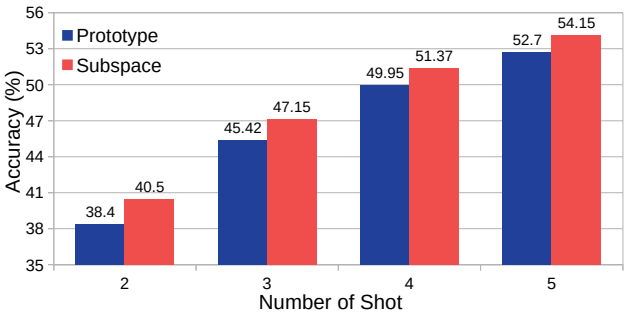


Figure 1: The accuracy of prototype and subspace classifiers evaluated with few (2-5) images. The feature extractor is ResNet-34 trained on the ImageNet. A prototype is an average pooling of few images within the same class and a subspace is the class specific basis vectors. The prototypes and subspaces are constructed directly from the generated features without additional learnable parameters.

1. Introduction

Various studies show that many deep learning techniques in computer vision, speech recognition and natural language understanding, to name but a few, will fail to produce reliable models that generalize well if limited annotations are available. Apart from the labor associated with annotating data, precise annotation can become ill-posed in some cases. One prime example of such a difficulty is object detection labeling which requires annotating bounding boxes of objects as explained in [1]. In some other cases, labeling process may require expert knowledge (*e.g.* sign language recognition [2]).

In contrast to the current trend in deep learning, humans can learn new objects from only a few examples. This in turn provides humans with lifelong learning abilities. Inspired by such learning abilities, several approaches are developed to study learning from limited samples [3–12]. This type of learning, known as Few-Shot Learning (FSL), has been tackled by a diverse set of ideas, from embedding learning [4, 13, 14], to adaptation techniques [7, 8] and even generative models [3, 15].

In this work, we first formulate FSL as a two-stage learning paradigm, namely, **1.** learning a universal feature extractor followed by **2.** learning to generate a classifier dynamically from limited data. We will demonstrate that many state-of-the-art FSL techniques fit nicely into such a learning paradigm. Furthermore, we will show that viewing the FSL as the above paradigm will be beneficial and provide us with tools to formalize FSL.

Once we establish the two-stage learning paradigm, we will turn our attention to how one can reliably generate a classifier from limited data. Aside from limited annotation, we will show that a requirement in many challenging FSL problems is to learn the classifier from high-dimensional data. This ultimately boils down to learning a symmetric function[1]

[1]A symmetric function is a function that has the same value given the arguments regardless of their order.

from high-dimensional data. To this end, we make another contribution and propose to construct the symmetric function using subspaces which have a long history in modeling visual data [16–19]. This differs in large from previous studies where the symmetric function is realized through a form of pooling (*e.g.*, averaging as in [20]).

As a motivating example, we compare and contrast the state-of-the-art prototypical networks [20] against our proposed subspace method using the CUB dataset [21]. To this end, for the universal feature extractor, we used the ResNet-34 trained on ImageNet [22]. We considered four FSL problems with various shots (two to five to be specific) and report the accuracy of the prototypical networks and our subspace method in Fig. 1. As will detail out shortly, in prototypical networks, one constructs low-shot classifiers by averaging all the samples within each class. Aside from being a natural choice, averaging is supported by **1.** In [23], it is shown that all symmetric functions over a set $\mathcal{X}$ can be written as $\rho(\sum_{x\in\mathcal{X}}\phi(x))$ for suitable transformations ρ and ϕ. **2.** In [11], authors note that the average of samples within a class is highly correlated with the parameters of classifiers learned by the softmax, hence one hopes that averaging reflects the true parameters of a class in FSL as well. Nevertheless, we observe that our subspace solution consistently and comfortably outperforms the prototypical networks. This compelling result along our thorough set of experiments on supervised and semi-supervised FSL (see for example Table 1 and 5) suggest that in few-shot regimes, there exists better ways to build classifiers from limited observations with subspace-based ones being our recommendation.

Contributions. In summary, we make the following contributions in this work:

i. Few-shot learning solutions are formulated within a framework of generating dynamic classifiers.

ii. We propose an extension of existing dynamic classifiers by using subspaces. We rely on a well-established concept stating that a second-order method generalizes better for classification tasks.

iii. We also introduce a discriminative formulation where maximum discrimination between subspaces is encouraged during training. This solution boosts the performance even further.

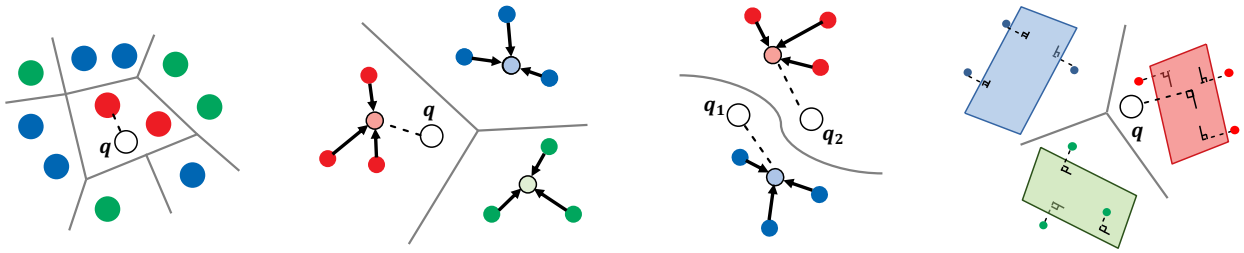
iv. We show that our method can make use of unlabeled data and hence it lends itself to the problem of semi-supervised few-shot learning and transductive setting. The robustness of such a variant is assessed in our experiments.

2. Related Work

In this section, we review the literature on few-shot learning and subspace methods for classification tasks. Few-shot learning was originally introduced to imitate the human learning ability. Some of the early works use generative models and similarity learning to capture the variation within parts and geometric configurations of objects [3, 15, 24]. These works use hand-crafted features to perform few-shot classification. Constellation model proposed in [15] takes into account the object parts for inference. The geometric structure of these parts helps discriminate between different objects. Furthermore, Torralba *et al*. [24] exploit similar features on visual objects but the model does not exploit the geometric structure. Another non-deep solution is the work by Lake *et al*. [3] which uses a set of primitives (strokes) to model few-shot classification. The above few-shot classification methods are not trained end-to-end and the given tasks are non-episodic.

The deep learning has been very successful in learning discriminative features from images. Santoro *et al*. [25] and Vinyals *et al*. [4] attempted to solve few-shot classification with end-to-end deep neural networks. In majority of cases, the network, trained from episodes, aims to infer the underlying discriminative model of specific tasks from limited data. Meta-learning can also be used to obtain fast adaptive networks. A prominent idea is to learn initial values for the parameters (weights) of the neural network. With proper initialization, one can expect the network to adapt to different tasks using backpropagation from limited samples. Sachin *et al*. [8] uses long-short term memory (LSTM) to embed the gradients w.r.t. a given task to train the network. MAML [7] does not use LSTM to encode the gradients but it can still perform meta-learning, usually with a better performance. As an extension, MAML++ [26] uses an importance scheme to weigh the loss during the gradient updates. MetaNets [27] is another fast adaptive network with a mix of so-called fast and fixed weights. The fast weights change through backpropagation while the fixed-weights do not change. Thus, one can see this method as an optimization applied to selected weights only.

FSL based on metric-learning is the closest direction to our work. Matching networks [4] and Siamese networks [13] learn sample-wise metric, meaning that distances to samples are used to determine the label of the query. In prototypical networks [20], Snell *et al*. extended the idea from samples to class-wise metric. The descriptors from all the samples of a specific class are grouped and considered as the class prototypes. The prototypes are subsequently used for inference. Learning a non-linear relationship between class representations and queries can be modeled by neural networks as shown for example in Relation Networks [14]. The underlying metric is learnt to preserve small distances between feature vectors sharing the same class label. Qiao *et*



(a) (b) (c) (d)

Figure 2: Various classifiers for few-shot classification. (a) Matching networks create pairwise classifiers. (b) Prototypical networks create mean classifiers based on the sample in the same class. (c) Relation networks produce non-linear classifiers. (d) Our proposed method creates classifiers using subspaces.

al. [11] observed that the activation of a network is correlated with weights of its classifier (final layer) and advocates that prototype made of the activation is sufficient for classification. Other works use feature attention modules [28, 29] to modulate features for few-shot learning [30, 31].

Several recent works target few-shot semi-supervised learning (FS-SSL). Garcia *et al*. [32] exploit graph neural networks for semi-supervised setting where unlabeled data is connected with the labeled data via Graph Neural Networks (GNN). Then, the features extracted from GNN are employed to classify the query. Another protocol for FS-SSL proposed by Ren *et al*. [33] shows that unlabeled images help samples from the support set to increase the performance of few-shot classification. The method proposed in [33] is based on the prototypical networks [20] with prototypes refined by the use of unlabeled images.

3. Problem Setting

We start by defining the terminology used in few-shot learning. A few of samples are trained for every iteration in meta-learning fashion. To obtain a trained model, so-called *episodes* are used to sample the data. An episode $\mathcal{T}_i$ consists of two sets, the support set S and the query set Q. This learning paradigm depicts how machine can improve their ability given fragmented data in each iteration. Specifically, the deep embeddings learn with limited amount of labels and inputs per episode. This learning paradigm is well-known as N-way K-shot classification (*e.g.*, 20-way 1-shot and 5-way 5-shot). We introduce our notations for the (N-way, K-shot) few-shot learning. Each episode or task $\mathcal{T}_i$ is composed of the support set $S = \{(\boldsymbol{x}_{1,1}, c_{1,1}), (\boldsymbol{x}_{1,2}, c_{1,2}), \cdots, (\boldsymbol{x}_{N,K}, c_{N,K})\}$ and the query set $Q = \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_{N\times M}\}$, where $\boldsymbol{x}_{i,j}$ denotes the j-th sample from class i and $c_{i,j} \in \{1, \cdots, N\}$. In the semi-supervised setting, there exist additionally an unlabeled set $\mathcal{R} = \{\boldsymbol{r}_1, .., \boldsymbol{r}_U\}$ within an episode.

A related problem is semi-supervised few-shot learning where unlabeled data is provided to the model. In the literature, various configurations are considered for semi-supervised few-shot learning *e.g.*, [32–34]. In this work, we follow the challenging protocol in [33] where the so-called *distractors* are introduced. Thus, an episode includes the support set S, query set Q, and unlabeled set $\mathcal{R}$. The support (labeled) S and query Q sets are configured as in few-shot learning. Additionally, an unlabeled set $\mathcal{R}$ is provided to assist the classification task within an episode. In the unlabeled set, there are samples from two different sources: the support classes and the *distractor* classes. As the name implies, samples from *distractor* classes are irrelevant to the classification task and represent classes outside the support set.

4. Proposed Method

4.1. Preliminary

We consider a few-shot learning problem in two stages: the feature extractor and the dynamic classifier. Let $f_\Theta : \mathcal{X} \rightarrow \mathbb{R}^D$ be a mapping from the input space $\mathcal{X}$ to a D-dimensional representation realized by a neural network and $\boldsymbol{X}_c = \{\boldsymbol{x}_{c,1}, \cdots, \boldsymbol{x}_{c,K}\}$ be a class-specific set. We formulate the problem of few-shot learning as generating dynamic classifiers. To this end, the final layer of a neural network along the softmax layer implements:

$$p(c|\boldsymbol{q}) = \frac{\exp(\boldsymbol{W}_c^\top f_\Theta(\boldsymbol{q}))}{\sum_{c'} \exp(\boldsymbol{W}_{c'}^\top f_\Theta(\boldsymbol{q}))} = \frac{\exp(d_c(\boldsymbol{q}))}{\sum_{c'} \exp(d_{c'}(\boldsymbol{q}))}, \quad (1)$$

where $\boldsymbol{W}_c$ is a weight of class c. Then, the problem of FSL can be understood as how $\boldsymbol{W}$ can be generated once a new task is provided. To showcase this setup, we discuss the pairwise classifier, the prototype, and the non-linear classifier below.

Pair-Wise Classifier. It is possible to build a classifier directly from samples by calculating the similarity between them as shown in Fig. 2 (a). One seminal work using this

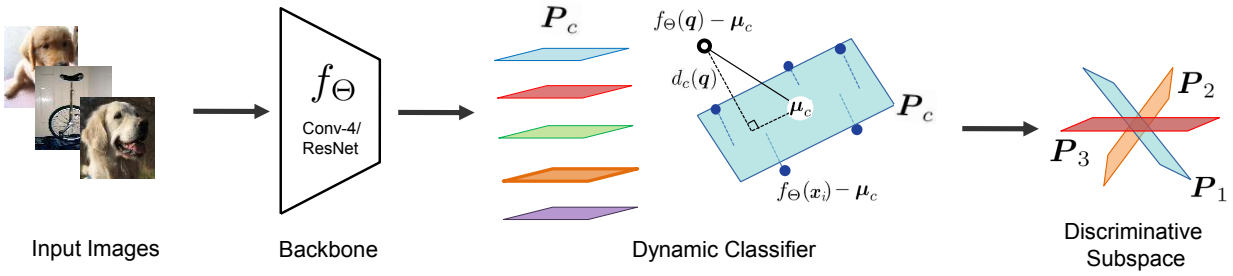


Figure 3: The overall pipeline of our approach. The subspace classifier replaces a classifier with a single vector per class. Discriminative method is then applied to maximize the margin between subspaces.

classifier is Matching Networks [4]. The samples are embedded through LSTMs and attention modules. However, this method does not poses an invariance w.r.t. the order of input images which affects the accuracy. The classifier weight $\boldsymbol{W}_c$ is substituted with a function $g(\cdot)$ (*e.g.* LSTMs) to encode the samples. Then the class specific samples are summarized and a cosine similarity used for prediction.

Prototype Classifier. Based on the observation for few-shot classification in [11], the parameters from the last fully-connected layer and prototypes correlate. Thus, the classifier is generated from the prototypes. By introducing a simple multi-layer perceptron, the average of feature vectors from the final activation layer is used to perform few-shot classification. This observation is also confirmed by prototypical networks [20] that learn directly feature embeddings. Some of the following works also use prototypes as dynamic classifiers such as [35, 36]. Thus, the $\boldsymbol{W}_c$ is substituted with $\frac{1}{K}\sum_{\boldsymbol{x}_i \in \boldsymbol{X}_c} f_\Theta(\boldsymbol{x}_i)$. Furthermore, this approach preserves the symmetric property (invariance to order of images) because the average operation is performed to generate the classifier. The illustration is depicted in Fig. 2 (b).

Non-Linear Binary Classifier. This approach exploits the non-linearity of the decision boundaries. Relational networks use a non-linear binary classifier to calculate the similarity as shown in Fig 2 (c). Let $\boldsymbol{z} = (f_\Theta(\boldsymbol{x}_i), f_\Theta(\boldsymbol{q}))$ and $\boldsymbol{M} \in \mathbb{R}^{2D}$ is the learnable classifier (comparator). We can redefine Eq. 1 as $p(c|\boldsymbol{q}) = \sigma(\boldsymbol{z}^\top \boldsymbol{M})$, where σ is a non-linear function (*e.g.*, sigmoid). Even though this classifier does not use a softmax function, it follows the principle of a generating classifier that learns a comparison of datapoint pairs.

4.2. Subspaces for Few-Shot Classification

We propose to model points by subspaces $\{\boldsymbol{Z}_i\}_{i=1}^N$. Each subspace $\boldsymbol{Z}_i$ has a basis represented by $\mathbb{R}^{D\times n} \ni \boldsymbol{B}_i = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n]; n \leq D$, with $\boldsymbol{B}_i^\top \boldsymbol{B}_i = \mathbf{I}_n$. Our goal is to learn the feature extractor Θ to generate subspaces, *ie.*, the function in a way that the resulting space is suitable for subspace classifiers.

A basis for the subspace representing class c can be obtained by a matrix decomposition *e.g.*, singular value decomposition (SVD). We emphasize that more involved techniques to obtain robust subspaces can potentially improve the algorithm. Nevertheless, our goal is to assess whether the concept of subspace modeling for few-shot learning is well justified, thus we opt for truncated SVD in our implementation.

4.3. Subspace Classifiers

High-order information is preferred than low-order to improve the capability of the classifier. A subspace method can form a robust classifier. Below, we describe how to create a subspace and classify based on it. A new set of samples encoded by Θ can be expressed as $\tilde{\boldsymbol{X}}_c = [f_{\Theta,}(\boldsymbol{x}_{c,1}) - \boldsymbol{\mu}_c, \cdots, f_\Theta(\boldsymbol{x}_{c,K}) - \boldsymbol{\mu}_c]$, where $\boldsymbol{\mu}_c = \frac{1}{K}\sum_{\boldsymbol{x}_i \in \boldsymbol{X}_c} f_\Theta(\boldsymbol{x}_i)$. One of the classification methods on a subspace is to find the closest distance between the datapoint to its projection onto the subspace. To this end, a class-specific projection matrix $\boldsymbol{P}_c$ is calculated from $\tilde{\boldsymbol{X}}_c$. Now a query $\boldsymbol{q}_j$ can be projected onto $\boldsymbol{P}_c$ and the classification based on the shortest distance from the query to its projection onto $\boldsymbol{P}_c$ (in original space) is performed. Our general subspace classifier is defined as:

$$d_c(\boldsymbol{q}) = -\|(\mathbf{I} - \boldsymbol{M}_c)(f_\Theta(\boldsymbol{q}) - \boldsymbol{\mu}_c)\|^2, \tag{2}$$

where $\boldsymbol{M}_j = \boldsymbol{P}_c \boldsymbol{P}_c^\top$ and $\boldsymbol{\mu}_c$ can be interpreted as the offset between a point and the subspace.Thus, $\boldsymbol{P}_c$ is a truncated matrix of a matrix $\boldsymbol{B}_c$ with orthogonal basis for the linear subspace spanning $\mathbb{X}_c = \{f_\Theta(\boldsymbol{x}_i);\ y_i = c\}$ (hence, $\boldsymbol{B}_c^\top \boldsymbol{B}_c = \mathbf{I}$).

We define the probability of the query assigned to class c using a softmax function as:

$$p_{c,q} = p(c|\boldsymbol{q}) = \frac{\exp\left(d_c(\boldsymbol{q})\right)}{\sum_{c'} \exp\left(d_{c'}(\boldsymbol{q})\right)}. \tag{3}$$

Now, we can minimize the negative log of Eq. 3 and update Θ. To train the whole framework, backpropagation through SVD is required which is available in modern deep learning packages such as PyTorch [37]. Hereafter, we call our proposed method as deep subspace networks (DSN).

4.4. Discriminative Deep Subspace Networks

Our goal in this part is to enhance DSN by learning representations that lead to more discriminative subspaces. In doing so, we make use of the Grassmannian geometry [38] and propose to maximize the distance between subspaces during training. This can be achieved with ease using the projection metric on Grassmannian which enjoys several useful properties (see [39]). To be more specific, given the basis of two subspaces $\boldsymbol{P}_3$ and $\boldsymbol{P}_j$, the projection metric is defined as:

$$\delta_p^2\left(\boldsymbol{P}_i, \boldsymbol{P}_j\right) = \left\| \boldsymbol{P}_i \boldsymbol{P}_i^\top - \boldsymbol{P}_j \boldsymbol{P}_j^\top \right\|_F^2 = 2n - 2\|\boldsymbol{P}_i^\top \boldsymbol{P}_j\|_F^2. \tag{4}$$

Maximizing the projection metric is achieved by minimizing $\|\boldsymbol{P}_i^\top \boldsymbol{P}_j\|_F^2$, yielding the following loss:

$$-\frac{1}{NM}\sum_c \log(p_{c,q}) + \lambda \sum_{i \neq j} \|\boldsymbol{P}_i^\top \boldsymbol{P}_j\|_F^2. \tag{5}$$

Algorithm 1 explains the steps of training DSN. Our overall pipeline is depicted in Fig. 3

Algorithm 1 Train Deep Subspace Networks

Input: Each episode $\mathcal{T}_i$ with S and Q

$\Theta_0 \leftarrow$ random initialization
for t in $\{\mathcal{T}_1, ..., \mathcal{T}_{N_\mathcal{T}}\}$ **do**
  for k in $\{1, ..., N\}$ **do**
    $\tilde{\boldsymbol{X}}_c \leftarrow \boldsymbol{S}_c$
    Calculate the average of the class
    Calculate mean refinement (MR) using Eq. 6
    Subtract $\tilde{\boldsymbol{X}}_c$ with an offset
    $[\mathcal{U}, \Sigma, \mathcal{V}^\top] \leftarrow$ Decompose$(\tilde{\boldsymbol{X}}_c)$
    $\boldsymbol{P}_c \leftarrow$ Truncate $\mathcal{U}_{1,...,n}$
    for $\boldsymbol{q}$ in Q **do**
      Compute $d_c(\boldsymbol{q})$ using Eq. 2
    end for
  end for
  Compute final loss $\mathcal{L}_t$ using Eq. 5
  Update Θ using $\nabla\mathcal{L}_t$
end for

4.5. DSN for Semi-Supervised Few-Shot Learning

In what follows, we extend the model developed in § 4.2 to address semi-supervised few-shot learning. In doing so, we need to take advantage of the unlabeled data to fit better subspaces to our data. We achieve this by refining the center of each class (mean-refinement) according to

$$\tilde{\boldsymbol{\mu}}_c = \frac{K\boldsymbol{\mu}_c + \sum_i m_i f_\Theta(\boldsymbol{r_i})}{K + \sum_i m_i}\ , \tag{6}$$

where,

$$m_i = \frac{\exp(-\|f_\Theta(\boldsymbol{r}_i) - \boldsymbol{\mu}_c)\|^2)}{\sum_{c'} \exp(-\|f_\Theta(\boldsymbol{r}_i) - \boldsymbol{\mu}_{c'})\|^2)}\ , \tag{7}$$

where m_i is the soft-assignment score for unlabeled samples. To work at the presence of *distractors*, we use a fake class with zero mean as in [33]. We empirically observed that such a simple modification to the means can improve the results without the need of refining the matrix decomposition step. Moreover, this technique is also applicable for transductive setting using query set as unlabeled data to refine mean of classes.

Remark 1. *To the best of our knowledge, subspaces have been used to address FSL in [40, 41] and our preliminary study [42]. A major difference between this work and TAPNET [40] is that the projection in our method is class-specific, while TAPNet makes use of task-specific projections. Our preliminary work [42] which precedes by ∼8 months the work of Devos and Grossglauser [41] share the same spirit and can be considered as a class specific subspace method for FSL.*

5. Experiments

Below we contrast and assess our method against state-of-the-art techniques on four challenging datasets, namely *mini*-ImageNet [8], *tiered*-ImageNet [33], CIFAR [43], and Open MIC [44]. Moreover, we used several CNN backbones such as 4-convolutional layers (Conv-4) as implemented in [20] and ResNet-12 as employed in [45] in our entire experiments for standard few-shot classification. We follow a general practice to evaluate the model with N-way K-shot and 15 query images. While perturbation analysis and semi-supervised few-shot (SS-FSL) classification, Conv-4 is adopted. The reported results of deep subspace networks (DSN) are provided on all datasets.

***mini*-ImageNet.** The *mini*-ImageNet [8] contains 60,000 images of the ImageNet [46] datasets. Images in the *mini*-ImageNet are of size 84×84 and represent 100 classes with 64, 16, and 20 classes used for training, validation, and testing, respectively. Every class has 600 images following the image list from [8]. It is clearly shown from previous work (*e.g.*, [47]) that CNN backbone affects the performance. Thus, we employ 4-convolutional layer (4-Conv) and ResNet-12 to make fair comparisons. We also use the *mini*-ImageNet for semi-supervised classification with 40% of labeled data.

***tiered*-ImageNet.** This dataset is also derived from ImageNet but contains a broader set of classes compared to the *mini*-ImageNet. There are 351 classes from 20 different categories for training, 97 classes from 6 different categories for validation, and 160 classes from 8 different categories for testing. We follow the implementation of 4-Conv and ResNet-12 backbones and image size of 84×84 as on *mini*-ImageNet.

Model	Backbone	1-shot	5-shot
Matching Nets [4]	Conv-4	43.56 ± 0.84	55.31 ± 0.73
MAML [7]	Conv-4	48.70 ± 1.84	63.11 ± 0.92
Reptile [48]	Conv-4	49.97 ± 0.32	65.99 ± 0.58
R2-D2 [49]	Conv-4	48.70 ± 0.60	65.50 ± 0.60
Prototypical Nets [20]	Conv-4	44.53 ± 0.76	65.77 ± 0.66
Relation Nets [14]	Conv-4	50.44 ± 0.82	65.32 ± 0.70
DSN	Conv-4	$\mathbf{51.78 \pm 0.96}$	$\mathbf{68.99 \pm 0.69}$
DSN-MR	Conv-4	$\mathbf{55.88 \pm 0.90}$	$\mathbf{70.50 \pm 0.68}$
Meta-Nets [27]	ResNet-12	57.10 ± 0.70	70.04 ± 0.63
SNAIL [10]	ResNet-12	55.71 ± 0.99	68.88 ± 0.92
AdaResNet [50]	ResNet-12	56.88 ± 0.62	71.94 ± 0.57
TADAM [51]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30
Prototypical Nets [20]	ResNet-12	59.25 ± 0.64	75.60 ± 0.48
FEAT [30]	ResNet-12	61.72 ± 0.11	78.32 ± 0.16
CTM [52]	ResNet-18	62.05 ± 0.55	78.63 ± 0.06
Qiao *et al.*‡ [11]	WRN-28-10	59.60 ± 0.41	73.74 ± 0.19
LwoF [36]	WRN-28-10	60.06 ± 0.14	76.39 ± 0.11
LEO‡ [53]	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12
wDAE-GNN‡ [54]	WRN-28-10	62.96 ± 0.15	78.85 ± 0.10
MetaOpt-SVM‡ [45]	ResNet-12	64.09 ± 0.62	80.00 ± 0.45
DSN	ResNet-12	$\mathbf{62.64 \pm 0.66}$	$\mathbf{78.83 \pm 0.45}$
DSN-MR	ResNet-12	$\mathbf{64.60 \pm 0.72}$	$\mathbf{79.51 \pm 0.50}$
DSN‡	ResNet-12	$\mathbf{65.38 \pm 0.63}$	$\mathbf{81.25 \pm 0.45}$
DSN-MR‡	ResNet-12	$\mathbf{67.09 \pm 0.68}$	$\mathbf{81.65 \pm 0.69}$

Table 1: Comparison with the state of the art. 5-way few-shot classification results with 95% confidence interval on *mini*-ImageNet dataset with various backbones for 1-shot and 5-shot. Methods with ‡ include training and validation sets for training the models.

CIFAR-100. We evaluate on the CIFAR-FS data split. All images on these datasets are 32×32 and the number of samples per class is 600. The CIFAR-FS dataset [49] is a few-shot learning benchmark containing all 100 classes from CIFAR-100 [43]. The dataset is divided into 64, 16 and 20 for training, validation, and testing, respectively.

Open MIC. This dataset [44] contains images from 10 museum exhibition spaces. In this dataset, there are 866 classes and 1-20 images per class. The images undergo various photometric and geometric distortions, the classes are often fine-grained in their nature, thus making few-shot learning problem challenging. The protocols and baselines we use are proposed in [55] but excludes the easiest to classify classes to make it possible for testing more than 1-shot then we rerun the SoSN [55] method. The dataset is divided into four subsets: *p1=(shn+hon+clv)*, *p2=(clk+gls+scl)*, *p3=(sci+nat)*, *p4=(shx+rlc)*. Protocol [55] assumes evaluations on *p1→p2*, *p2→p3*, *p3→p4*, and *p4→p1*, where *x→y* denotes training on subset *x* and testing on subset *y*. Training in a subset and testing in another subset depicts a few-shot learning problem because the objects in every museum are distinct with different backgrounds. Note that, we eliminate classes with less than 3 examples and rerun all algorithms in our experiment.

Model	Backbone	1-Shot	5-Shot
Prototypical Nets [20]	ResNet-12	61.74 ± 0.77	80.00 ± 0.55
CTM [52]	ResNet-18	64.78 ± 0.11	81.05 ± 0.52
LEO‡ [53]	WRN-28-10	66.33 ± 0.05	81.44 ± 0.09
MetaOpt - SVM‡ [45]	ResNet-12	65.81 ± 0.74	81.75 ± 0.53
DSN	ResNet-12	$\mathbf{66.22 \pm 0.75}$	$\mathbf{82.79 \pm 0.48}$
DSN-MR	ResNet-12	$\mathbf{67.39 \pm 0.82}$	$\mathbf{82.85 \pm 0.56}$
DSN‡	ResNet-12	$\mathbf{66.83 \pm 0.73}$	$\mathbf{83.31 \pm 0.64}$
DSN-MR‡	ResNet-12	$\mathbf{68.44 \pm 0.77}$	$\mathbf{83.32 \pm 0.66}$

Table 2: 5-way few-shot classification results on *tiered*-ImageNet with 95% confidence intervals. Methods with ‡ include training and validation sets for training the models.

Model	1-Shot	5-Shot
Prototypical Nets [4]	72.2 ± 0.7	83.5 ± 0.5
MetaOpt - RR [45]	72.6 ± 0.7	84.3 ± 0.5
MetaOpt - SVM [45]	72.0 ± 0.7	84.2 ± 0.5
MetaOpt - SVM‡ [45]	72.8 ± 0.7	85.0 ± 0.5
DSN	$\mathbf{72.3 \pm 0.8}$	$\mathbf{85.1 \pm 0.6}$
DSN-MR	$\mathbf{75.6 \pm 0.9}$	$\mathbf{86.2 \pm 0.6}$
DSN‡	$\mathbf{73.6 \pm 0.9}$	$\mathbf{86.3 \pm 0.6}$
DSN-MR‡	$\mathbf{78.0 \pm 0.9}$	$\mathbf{87.3 \pm 0.6}$

Table 3: 5-way few-shot classification results on the CIFAR-FS dataset using ResNet-12 with 95% confidence intervals. Methods with ‡ include training and validation sets for training the models.

5.1. Few-shot Learning

We follow the general practice and evaluate our method on *mini*-ImageNet, *tiered*-ImageNet, CIFAR-FS, and Open MIC when it comes to few-shot learning and classification. The CNN architectures for *mini*-ImageNet are the same as the one used in [47] with 4 convolutional layers (Conv-4) and ResNet-12 [56]. While, only ResNet-12 is used for CIFAR-FS and *tiered*-ImageNet. We use ADAM [57] for optimizing Conv-4 and SGD for optimizing ResNet-12. For a fair comparison, we conduct similar experimental setups. Conv-4 backbone is trained without data augmentation following the other methods and cut learning rate to half every 5K episodes. We trained on 5-way 1-shot and 5-shot, then applied the same classification task setup during testing for Conv-4. Note that, Prototypical Nets [20] using Conv-4 were also trained and tested on 5-way. The training for ResNet-12 is performed with data augmentation and the learning rate is set 0.1 initially then it is adjusted to 0.003, 0.00032, and 0.00014 at epochs 12, 30, and 45, respectively. Moreover, the training strategy in [45] is utilized with 15-shot, 10 query images, and 8 episodes per batch. We cross-validated from a validation set and set $\lambda = 0.03$ for all experiments. The accuracy is evaluated over 1000 episodes.

Model	5-way 1-shot					5-way 3-shot				
	$p1 \rightarrow p2$	$p2 \rightarrow p3$	$p3 \rightarrow p4$	$p4 \rightarrow p1$	Avg	$p1 \rightarrow p2$	$p2 \rightarrow p3$	$p3 \rightarrow p4$	$p4 \rightarrow p1$	Avg
Matching Nets [4]	69.40	57.30	76.35	53.68	64.18	84.10	74.20	87.47	70.83	79.15
Relation Nets [14]	70.10	49.70	66.90	46.90	58.40	80.90	61.90	78.50	58.90	70.05
Prototypical Nets [20]	66.33	52.03	74.28	54.30	61.74	81.60	73.55	83.55	69.15	76.96
SoSN [55]	**78.00**	60.10	75.50	57.80	67.85	87.10	72.60	85.90	72.80	79.60
DSN	**75.87**	**62.13**	**78.25**	**62.11**	**69.59**	**87.93**	**75.78**	**88.42**	**76.59**	**82.18**

Table 4: Few-shot classification results using Conv-4 on the Open MIC dataset for 5-way 1-shot and 3-shot.

Dataset	Model	1-shot		5-shot	
		w/o D	w/ D	w/o D	w/ D
***mini*-ImageNet**	PN-SSL, Non-Masked [33]	50.09 ± 0.45	48.70 ± 0.32	64.59 ± 0.28	63.55 ± 0.28
	PN-SSL, Masked [33]	50.41 ± 0.31	49.04 ± 0.31	64.39 ± 0.24	62.96 ± 0.14
	Semi DSN	$\mathbf{53.01 \pm 0.82}$	$\mathbf{51.01 \pm 0.78}$	$\mathbf{69.12 \pm 0.62}$	$\mathbf{67.12 \pm 0.81}$
***tiered*-ImageNet**	PN-SSL, Non-Masked [33]	51.85 ± 0.25	51.36 ± 0.31	$70.25 \pm 0.31\%$	68.32 ± 0.22
	PN-SSL, Masked [33]	52.39 ± 0.44	51.38 ± 0.38	$69.88 \pm 0.20\%$	69.08 ± 0.25
	Semi DSN	$\mathbf{54.06 \pm 0.96}$	$\mathbf{53.89 \pm 0.83}$	$\mathbf{72.07 \pm 0.69}$	$\mathbf{70.15 \pm 0.81}$

Table 5: 5-way semi-supervised few-shot classification results using Conv-4 on *mini*-ImageNet and *tiered*-ImageNet with 40% and 10% labeled data, respectively. We show the classification results with (w/ D) and without *distractors* (w/o D).

By design, our method needs more than one sample to identify the span of a subspace. Thus, for 1-shot case, we generate an additional sample by data augmentation through flipping support images.

Results. Below, we provide our results based on the Conv-4 and ResNet-12 for a comprehensive comparison. Note that different backbones can affect the performance of few-shot learning. For the *mini*-ImageNet, Table 1 shows that our method outperforms state-of-the-art methods with various CNN backbones and the number of samples for 5-way 5-shot and 1-shot. Our method can also benefit from the mean refinement (MR) of the query set. Our method is even better on deeper CNN with more parameters such as ResNet-12 [56]. Our performance is 1.3% better than MetaOpt-SVM [45] on 5-way 1-shot and 5-shot. Our method also consistently outperforms the other methods the *tiered*-ImageNet and CIFAR-FS datasets (see Tables 2 and 3).

On the Open MIC dataset (see Table 4), a similar trend can be observed. Our methods outperform state-of-the-art embedding methods for few-shot learning (*ie*., Matching Nets [4], Prototypical Nets [20], and Second-order Similarity Network (SoSN) [55]). The results show that our subspace representation is robust to various photometric and geometric distortions posed by the Open MIC dataset, and it can model fine-grained concepts contained in this dataset well. Open MIC contains different exhibitions with different types of objects. Our model can generalize to different subsets of objects on Open MIC with around 2% gain compared to other methods.

5.2. Semi-Supervised Few-shot Learning

For experiments in this section, we used the embedding architecture with 4-convolutional layers as in [33]. We followed the experimental setup proposed by [33]. The episode composition of labeled part of support and query sets is similar to the few-shot learning classification task, however, there is an additional unlabeled set provided in each episode. Our model was trained on 100K episodes on *mini*-ImageNet and *tiered*-ImageNet with 40% and 10% of labeled data, respectively. We used the ADAM solver [57], then set the learning rate to 0.001 with the weight decay and cut the rate to half every 10K episodes.

The training was performed in the semi-supervised setting for which the unlabeled set was also used. The unlabeled set was composed of the samples from the classes in the support set and *distractor* classes. The number of supporting classes and *distractor* classes were set to five for training and testing. In the training stage, the number of samples in the unlabeled set was 50 (five samples from each class). In the testing stage, the unlabeled set consisted of 20 samples from each class. The query set had 20 samples per class for testing purposes. λ was set to 0.03 and 0.005 for semi-supervised few-shot learning on *mini*-ImageNet and *tiered*-ImageNet, respectively.

Results. The accuracy is evaluated over 600 episodes. The results are averaged over 10 random splits of labeled and unlabeled sets. The *semi-supervised* experiment detailed in Table 5 shows that our method improves the performance by exploiting unlabeled data. Our results are compared to proto-

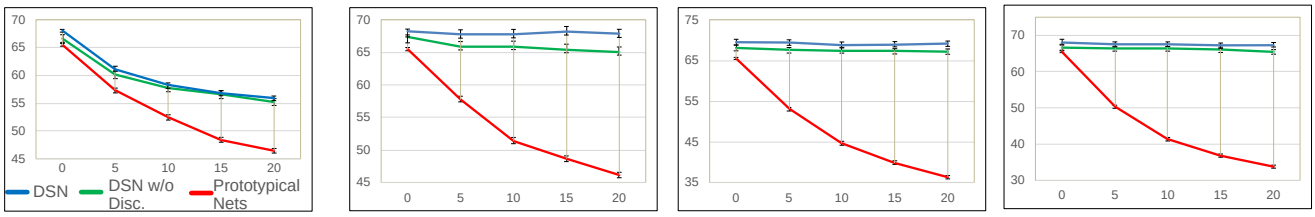


Figure 4: Experiments in the presence of outliers and additive noise on *mini*-ImageNet for 5-way 5-shot using Conv-4. The results of DSN, DSN without discriminative term, and prototypical networks are shown (see the legend). The *first column* shows the impact of introducing outliers among support samples (the classes of outliers are disjoint with the support classes of samples). The *second, third and fourth* columns show the impact of introducing noisy samples generated randomly according to the Gaussian distribution with random means and variance of $\sigma = \{0.15, 0.3, 0.4\}$, respectively. The performance is measured w.r.t. the increasing number of outliers and noisy samples (x-axis).

Approach	5-way 1-shot	5-way 5-shot
Without Disc. Term	50.44 ± 0.88	67.22 ± 0.69
With Disc. Term	51.78 ± 0.96	68.99 ± 0.69

Table 6: Few-shot classification accuracy for DSN using Conv-4 with and without the discriminative term on *mini*-ImageNet.

typical networks on semi-supervised learning (SS-FSL) with soft K-means (non-masked) and masked K-means (masked), as proposed by [33].

5.3. Ablation Study

Discriminative Term. Below, an ablation study w.r.t. the discriminative term is performed. The discriminative term in Eq. 4 encourages the orthogonality between subspaces of different classes. This term leads to a performance boost on few-shot classification tasks. We investigated results for this mechanism in Table 6 given the Conv-4 backbone. From results we conclude that the network learns discriminative subspaces which are pushed away from each other. This empirical study proves that the discriminative term gives a performance boost and results in more discriminative subspaces for classification.

Subspace Dimensionality. In comparison to other models such as matching networks, prototypical networks, and relation networks, our DSN comes with an additional hyper-parameter, the dimensionality of the subspaces (*ie*., n). As a rule of thumb, we recommend to use $n = K - 1$ to train and test our model. In fact, DSN exhibits a large degree of robustness to n, which in turns, makes training of our model simple. We observe that the choice of n from 2 to $K - 1$ does not affect the performance significantly ($\pm$ 0.5%) on *mini*-ImageNet using Conv-4 backbone.

6. Discussion

Robustness to Perturbations. One may argue whether noise poses problems in few-shot learning. However, some noise patterns might not be obvious when collecting the data. Thus, the data cannot be guaranteed to be free from noise. We observed in our experiments that the performance for standard methods degrades significantly with a small degree of perturbations added to signal, as depicted in Fig. 4. However, our subspace-based model handles such a noise well.

Computational Complexity. The computational complexity of our DSN approach is $\mathcal{O}(\min(ND^2K, NDK^2))$, where K, N, and D are the number of shot, way, and the feature dimensionality, respectively. Compared to the complexity of the prototypical networks approach, *ie*., $\mathcal{O}(NDK)$, our method is somewhat slower due to the use of the SVD step. However, to address the complexity of SVD, fast approximate SVD algorithms can be used [58].

7. Conclusions

This paper presents the DSN, a novel few-shot learning approach that employs a few-shot learning model via affine subspaces. Empirically, we showed that the representations learned via DSN are expressive across a wide-range of supervised and semi-supervised few-shot problems. Both of them are trained in meta-learning and the test set is not seen previously while training the model. The subspace model is proven to improve existing models by a large margin due to its nature to represent a few datapoints on a subspace.

In DSN, each class classifier is represented by the subspace formed by all its samples, meaning that each class is modeled by the span of its training datapoints. We showed that DSN is robust to noise in few-shot learning. Our experiments demonstrated that a higher classification accuracy can be obtained by simply encouraging subspaces to be separated from each other.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2189–2202, 2012.

[2] D. Li, C. Rodriguez, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1459–1469.

[3] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, pp. 1332–1338, 2015.

[4] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, 2016.

[5] E. Triantafillou, R. Zemel, and R. Urtasun, "Few-shot learning through an information retrieval lens," in *Advances in Neural Information Processing Systems*, 2017.

[6] Z. Xu, L. Zhu, and Y. Yang, "Few-shot object recognition from machine-labeled web images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[7] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, 2017.

[8] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2017.

[9] Y.-X. Wang, R. Girshick, M. Herbert, and B. Hariharan, "Low-shot learning from imaginary data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[10] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *International Conference on Learning Representations*, 2018.

[11] S. Qiao, C. Liu, W. Shen, and A. L. Yuille, "Few-shot image recognition by predicting parameters from activations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] J. O. Neill and P. Buitelaar, "Few shot transfer learning betweenword relatedness and similarity tasks using a gated recurrent siamese network," in *AAAI Conference on Artificial Intelligence*, 2018.

[13] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *International Conference on Machine Learning Deep Learning 2015 Workshop*, 2015.

[14] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[15] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 594–611, 2006.

[16] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[17] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 218–233, 2003.

[18] P. Zhou, Y. Hou, and J. Feng, "Deep adversarial subspace clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1596–1604.

[19] J. Wang and A. Cherian, "Gods: Generalized one-class discriminative subspaces for anomaly detection," in *The IEEE International Conference on Computer Vision*, October 2019.

[20] J. Snell, K. Swersky, and Z. Richard, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017.

[21] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012.

[23] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Advances in neural information processing systems*, 2017, pp. 3391–3401.

[24] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 854–869, 2007.

[25] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*, 2016, pp. 1842–1850.

[26] A. Antoniou, H. Edwards, and A. Storkey, "How to train your maml," in *International Conference on Learning Representations*, 2019.

[27] T. Munkhdalai and H. Yu, "Meta networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2554–2563.

[28] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 090–10 100.

[29] P. Fang, J. Zhou, S. K. Roy, L. Petersson, and M. Harandi, "Bilinear attention networks for person retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8030–8039.

[30] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Learning embedding adaptation for few-shot learning," *arXiv preprint arXiv:1812.03664*, 2018.

[31] R. Hou, H. Chang, M. Bingpeng, S. Shan, and X. Chen, "Cross attention network for few-shot classification," in *Advances in Neural Information Processing Systems*, 2019, pp. 4005–4016.

[32] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *International Conference on Learning Representations*, 2018.

[33] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *International Conference on Learning Representations*, 2018.

[34] R. Boney and A. Ilin, "Semi-supervised few-shot learning with prototypical networks," *arXiv preprint arXiv:1711.10856*, 2017.

[35] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[36] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4367–4375.

[37] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS Autodiff Workshop*, 2017.

[38] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM journal on Matrix Analysis and Applications*, vol. 20, pp. 303–353, 1998.

[39] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson, "Extrinsic methods for coding and dictionary learning on grassmann manifolds," *International Journal of Computer Vision*, vol. 114, pp. 113–136, 2015.

[40] S. W. Yoon, J. Seo, and J. Moon, "Tapnet: Neural network augmented with task-adaptive projection for few-shot learning," in *International Conference on Machine Learning*, 2019.

[41] A. Devos and M. Grossglauser, "Subspace networks for few-shot classification," *arXiv:1905.13613*, 2019.

[42] C. Simon, P. Koniusz, and M. Harandi, "Projective subspace networks for few-shot learning," *OpenReview, https://openreview.net/forum?id=rkzfuiA9F7*, 2018.

[43] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[44] P. Koniusz, Y. Tas, H. Zhang, M. Harandi, F. Porikli, and R. Zhang, "Museum exhibit identification challenge for the supervised domain adaptation and beyond," in *The European Conference on Computer Vision*, 2018.

[45] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 657–10 665.

[46] O. Russakovsky, J. Deng, H. Su, J. K. andSanjeev Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and e. a. Michael Bernstein, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.

[47] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *International Conference on Learning Representations*, 2019.

[48] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[49] L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *International Conference on Learning Representations*, 2019.

[50] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler, "Rapid adaptation with conditionally shifted neurons," in *International Conference on Machine Learning*, 2018, pp. 3661–3670.

[51] B. Oreshkin, P. Rodríguez López, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 719–728.

[52] H. Li, D. Eigen, S. Dodge, M. Zeiler, and X. Wang, "Finding task-relevant features for few-shot learning by category traversal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1–10.

[53] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," in *International Conference on Learning Representations*, 2019.

[54] S. Gidaris and N. Komodakis, "Generating classification weights with GNN denoising autoencoders for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[55] H. Zhang and P. Koniusz, "Power normalizing second-order similarity network for few-shot learning," in *Winter Conference on Applications of Computer Vision*, 2019.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[57] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[58] A. K. Menon and C. Elkan, "Fast algorithms for approximating the singular value decomposition," *ACM Trans. Knowl. Discov. Data*, vol. 5, pp. 13:1–13:36, 2011.