Deep Subspace Networks 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

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 introduce the *Deep Subspace Networks (DSN)*, a deep learning paradigm that learns embeddings from limited supervision. In contrast to previous studies, the embedding in DSN deems samples of a given class to form an affine subspace. We will empirically show that such modelling leads to robustness against perturbations (outliers and noises) and yields competitive results on the task of supervised few-shot classification.

1 Introduction

Supervised learning with deep architectures, though achieving remarkable results in many areas, requires large amount of annotated data. Few-Shot Learning (FSL), an emerging learning paradigm, tackles adapting models rapidly in the presence of limited data. The diverse ideas in this context include embedding features through metric learning [1, 2, 3], optimization technique [4, 5], and generative models [6, 7].

In this work, we propose a deep model that learns new concepts from limited data to address a challenging problem, namely few-shot classification. The goal of few-shot classification is to learn a model that can discriminate a given query by comparing it to a few of samples (the support set).

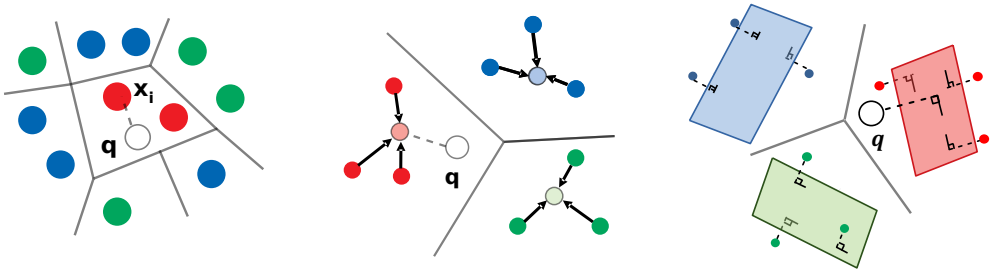
Our method, coined Deep Subspace Networks or DSN for short, models classifiers using low-dimensional affine subspaces. The use of subspaces to model images and sets has a long history in computer vision and machine learning. For example, it has been proved that the set of all reflectance functions (the mapping from surface normals to intensities) produced by Lambertian objects lie close to a low-dimensional linear subspace [8]. This makes our paper distinct and novel as compared to former studies, [2, 9, 3]. Fig. 1 provides a conceptual illustration of previous works and our approach.

We empirically observed that classifiers tailored towards capturing the structure of each class through low-dimensional affine subspaces could lead to robust models. Interestingly, such models can be built with minimum overheads and without opting for advanced methods in subspace creation (such as the notion of sparsity). In our experiments, subspaces for few-shot learning are less sensitive to perturbations such as outliers and noise compared to a previous technique.

Our contributions in this work are:

i. Few-shot learning is formulated as a classifier problem through subspaces. We rely on a well-established concept stating that samples of a class (and hence variations such as pose and illumination) can be effectively captured by low-dimensional affine spaces [10, 11].

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.



(a) (b) (c)

Figure 1: Feature embedding in (a) Matching Networks [2], (b) Prototypical Networks [9], and (c) our DSN method.

ii. We will show that the DSN is more robust to two forms of perturbations, namely outliers and noise contamination.

2 Deep Subspace Networks

For classification and in majority of cases, the final layer of a Deep Neural Network (DNN) is a softmax layer that calculates to what degree the input belongs to a particular class as

$$p(c|\boldsymbol{q}) = \frac{\exp(\boldsymbol{w}_c^\top \boldsymbol{q})}{\sum_{c'} \exp(\boldsymbol{w}_{c'}^\top \boldsymbol{q})} = \frac{\exp(s_c(\boldsymbol{q}))}{\sum_{c'} \exp(s_{c'}(\boldsymbol{q}))}. \tag{1}$$

In essence, one measures the similarity between the query sample $\boldsymbol{q}$ and a class c using the function $s_c(\cdot)$. In a softmax classifier, $s_c(\cdot)$ is identified as the inner product between the query and the class representative $\boldsymbol{w}_c$.

In FSL, one needs to identify the classifier parameters, $\boldsymbol{w}_c, \forall c$, from limited data. A prominent and natural idea here is to define $\boldsymbol{w}_c$ as the average of samples in class c. That is, $\boldsymbol{w}_c = \mathbb{E}_{\boldsymbol{x}\sim p_c}(\boldsymbol{x})$ as done in [9]. More involved methods benefit from auxiliary networks to transform the class representative (*i.e*, $\mathbb{E}_{\boldsymbol{x}\sim p_c}(\boldsymbol{x})$) to the parameters of the classifier [12].

This school of thought comes with an obvious drawback. Averages, as class representatives, are sensitive to perturbations and discard valuable information that might be beneficial for classification. In general, models that make use of higher-order statistics should be advantageous over the ones that only consider the first-order moments. However, with limited data, accurately estimating higher-order statistics is prone to errors, if not impossible, especially when high-dimensional spaces is considered.

To enrich our models and benefit from higher-order statistics while being vigilant to the nature and requirements of FSL problems, we propose to model low-shot classes with affine subspaces. Consider an (N-way, k-shot) FSL problem. In episodic learning [2, 13], an episode or task $\mathcal{T}_i$ is composed of a support set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N\times k}$ and a query set $\mathcal{Q} = \{(\boldsymbol{q}_j, y_j^q)\}_{j=1}^{m}$ with $\boldsymbol{x}_i, \boldsymbol{q}_j \in \mathcal{X}$ and $y_i, y_j^q \in \{1, 2, \cdots, N\}$. Our goal is to train a DNN, parameterized by Θ to realize a mapping $f_\Theta : \mathcal{X} \to \mathbb{R}^D$ such that classifiers defined using $\mathcal{S}$ perform well on $\mathcal{Q}$ over a large enough set of tasks $\mathcal{T}_i$. In doing so, we propose the following similarity function;

$$s_c(\boldsymbol{q}) = -\big\|f_\Theta(\boldsymbol{q}) - \pi_c(\boldsymbol{q})\big\|_2^2, \tag{2}$$

$$\pi_c(\boldsymbol{q}) = \boldsymbol{W}_c \boldsymbol{W}_c^\top f_\Theta(\boldsymbol{q}) - \boldsymbol{b}_c. \tag{3}$$

Here, $\boldsymbol{W}_c$ is an orthogonal basis for the linear subspace spanning $\mathbb{X}_c = \{f_\Theta(\boldsymbol{x}_i);\ y_i = c\}$ (hence, $\boldsymbol{W}_c^\top \boldsymbol{W}_c = \mathbf{I}$) and $\boldsymbol{b}_c = \mathbb{E}_{\boldsymbol{x}\sim p_c}\big(f_\Theta(\boldsymbol{x})\big)$ is a class-dependent bias term. In essence, $s_c(\boldsymbol{q})$ in Eq. 2 is the negative distance between $f_\Theta(\boldsymbol{q})$ and the affine subspace that spans the class c. To obtain $\boldsymbol{W}_c$ in Eq. 2, one can readily stack samples in $\mathbb{X}_c$ into columns of a matrix, followed by computing the left singular vectors of the resulting matrix. The full pseudocode to train DSN is written in Algorithm 1. To train the DSN, backpropagation through SVD is required which is available in modern deep learning packages such as PyTorch [14].

Algorithm 1 Train Deep Subspace Networks

Input: Each episode $\mathcal{T}_i$ with $S = \{(\boldsymbol{x}_{1,1}, c_{1,1}), \cdots, (\boldsymbol{x}_{N,K}, c_{N,K})\}$ and $Q = \{\boldsymbol{q}_1, ..., \boldsymbol{q}_{N\times M}\}$

$\Theta_0 \leftarrow$ random initialization
for t in $\{\mathcal{T}_1, ..., \mathcal{T}_{N_\mathcal{T}}\}$ **do**
  $\mathcal{L}_t \leftarrow 0$
  for k in $\{1, ..., N\}$ **do**
    $\boldsymbol{X} \leftarrow \boldsymbol{S}_c$ ▷ Get examples in the support set from class c
    $\boldsymbol{\mu}_c \leftarrow \frac{1}{K}\sum_{x\in\boldsymbol{X}} f_\Theta(\boldsymbol{x})$ ▷ Mean from the support set
    $\tilde{\boldsymbol{X}} \leftarrow [\boldsymbol{x}_i - \boldsymbol{\mu}_c, ..., \boldsymbol{x}_K - \boldsymbol{\mu}_c]$
    $[\mathcal{U}, \Sigma, \mathcal{V}^\top] \leftarrow \text{SVD}(\tilde{\boldsymbol{X}})$ ▷ Matrix factorization using SVD
    $\boldsymbol{W}_c \leftarrow \mathcal{U}_{1,...,n}$ ▷ Truncate the matrix
    for $\boldsymbol{q_j}$ in Q_k **do**
      Project $\boldsymbol{q_j}$ using Eq. 3
      Calculate $p_{j,k}$ from Eq. 2 with softmax
    end for
    $\mathcal{L}_t \leftarrow \frac{1}{N^2M}\sum_k\sum_j -\log\left(p_{j,k}\right)$
    Update Θ using $\nabla_\Theta\mathcal{L}_t$
  end for

Model	1-shot	5-shot
Matching Networks [2]	52.91 ± 0.88	68.88 ± 0.69
Prototypical Networks [9]	54.16 ± 0.82	73.68 ± 0.65
Relation Networks [3]	52.48 ± 0.86	69.83 ± 0.68
CTM [15] (fine-tune)	62.05 ± 0.55	78.63 ± 0.06
DSN	$\mathbf{56.32 \pm 0.79}$	$\mathbf{75.49 \pm 0.62}$
DSN (fine-tune)	$\mathbf{62.58 \pm 0.80}$	$\mathbf{79.62 \pm 0.71}$

Table 1: Comparison with the state-of-the-arts. 5-way 1-shot and 5-way 5-shot few-shot classification using ResNet-18 on the *mini*-ImageNet dataset.

3 Experimental Results

In this section, we assess our method against state-of-the-art techniques on the *mini*-ImageNet [5] and the Open MIC [16]. The *mini*-ImageNet [5] is a subset of ImageNet [17] that has 64, 16, and 20 classes for training, validation, and testing, respectively. The Open-MIC split following [18] contains images from 10 museum exhibitions with 866 classes and 1-20 images per class.

In our experiments, we used two backbones, namely ResNet-18 per [19] and 4-convolutional layers following [9] for the *mini*-ImageNet and the Open MIC, respectively. The experiments on both dataset show that the DSN performance is superior than previous state-of-the-arts. In all our experiments, we set the subspace dimension (n) as k-1 with k being the number of shots (samples per class). Note that, in case of 1-shot, we employed an image flipping to create a subspace.

In Table 1 and 2, we compare DSN against the matching networks [2], prototypical networks [9], and relation nets [3] using the 4 convolutional blocks as the backbone. As alluded to earlier, prototypical networks can be understood as a method to generate classifiers from first-order statistics, hence idea-wise the most important baseline here. Matching-networks does not benefit from statistical information and can be understood as a lazy classifier (pair-wise comparisons used towards classification). Superiority of DSN and prototypical networks over the matching network suggests the importance of benefiting from statistical information in designing the classifier. Relation-nets can be understood as a ranking-based classifier, different in nature but considered here for completeness.

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 [2]	69.4	57.3	76.4	53.7	64.2	84.1	74.2	87.5	70.8	79.2
Prototypical Networks [9]	66.3	52.0	74.3	54.3	61.8	81.6	73.6	83.6	69.2	77.0
Relation Networks [3]	70.1	49.7	66.9	46.9	58.4	80.9	61.9	78.5	58.9	70.1
DSN	**72.9**	**57.6**	**77.6**	**61.3**	**67.4**	**86.1**	**75.6**	**84.5**	**72.1**	**79.6**

Table 2: 5-way 1-shot and 5-way 3-shot few-shot classification accuracy on the Open MIC dataset using 4 convolutional blocks.

Interestingly and without making use of a complicated machinery, DSN can compete and even outperform state-of-the-art solutions when equipped with the ResNet-18 in comparison to CTM.

3.1 Robustness to Perturbations

Subspaces exhibit some degree of robustness in the presence of perturbations [20]. To empirically verify this, we assess the sensitivity of DSN by introducing two types of perturbations at the test time to already trained models. To this end, we randomly sampled data points from classes not presented in the support sets and included them in the support samples (robustness to outliers). Secondly, noisy samples are generated randomly according to a multivariate Gaussian distribution with random mean and variance of $\sigma = \{0.3, 0.4\}$, respectively. The results of this study are presented in Fig. 2. To summarize, our experiments show that both subspace and prototype methods are affected by outliers negatively. That said, our method exhibits a much better degree of resilience to outliers and noises.

In this section, we study the effect of perturbation on the performance of Prototypical Networks and DSN. More specifically, we considered the problems of 5-way 5-shot and 5-way 10-shot learning and introduced two types of perturbations at the test time with the trained models. Note that, the model is obtained from 5-way 5-shot training without perturbations.

Firstly, we randomly sampled examples from classes not presented in the support sets. Secondly, additive noise is generated randomly using a multivariate Gaussian distribution with random mean and variance of $\sigma = \{0.3, 0.4\}$. Both examples in these two types are included in the support examples for prototypes and subspaces creation.

The results of this study are depicted in Fig. 2. To summarize, our experiments show that both DSN and prototypical nets are affected by outliers negatively. That said, the DSN exhibits a much better degree of resilience to outliers. For example, modelling with prototypical nets leads to a drop of 19% and 12% percentage points when each support set has 20 outliers for the problem of 5-way 5-shot and 5-way 10-shot respectively. For the same experiment, the DSN modelling only suffers 11% and 9% percentage points of performance drop. When additive noise is considered, DSN behaves robustly for a wide-range of contamination. In contrast, the performance of prototypical nets drops rapidly and significantly in the presence of noise, reinforcing our idea that subspaces form indeed a more robust model for the task in hand.

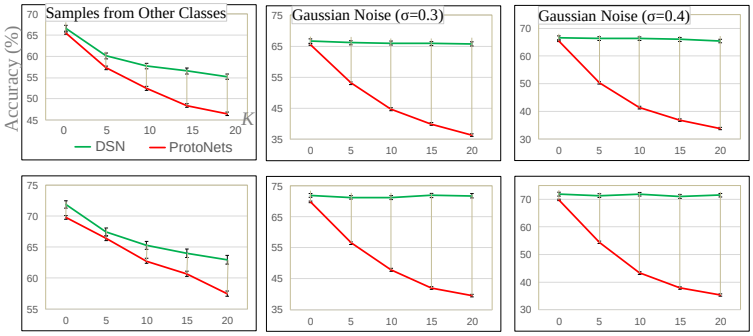


Figure 2: The plots show 5-way 5-shot (top) and 5-way 10-shot (bottom) results of the DSN and prototypical nets in the presence of outliers and additive noise. The performance is measured with increasing number of outliers and noisy examples (X-axes).

4 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 on 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. We showed that DSN is robust to perturbations compared to the other embedding approaches for few-shot learning. Moreover, we proposed a discriminative term to create subspaces such that the distance from two different subspaces representing different classes is maximized.

References

[1] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning Deep Learning 2015 Workshop*, 2015.

[2] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems(NIPS)*, 2016.

[3] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning(ICML)*, 2017.

[5] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations(ICLR)*, 2017.

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, 28:594–611, 2006.

[7] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350:1332–1338, 2015.

[8] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, 25:218–233, 2003.

[9] Jake Snell, Kevin Swersky, and Zemel Richard. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems(NIPS)*, 2017.

[10] Tae-Kyun Kim, Josef Kittler, and Roberto Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1005–1018, 2007.

[11] Ognjen Arandjelović and Roberto Cipolla. A pose-wise linear illumination manifold model for face recognition using video. *Computer vision and image understanding*, 113:113–125, 2009.

[12] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. Few-shot image recognition by predicting parameters from activations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[13] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

[14] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Autodiff Workshop*, 2017.

[15] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, 2019.

[16] Piotr Koniusz, Yusuf Tas, Hongguang Zhang, Mehrtash Harandi, Fatih Porikli, and Rui Zhang. Museum exhibit identification challenge for the supervised domain adaptation and beyond. In *The European Conference on Computer Vision (ECCV)*, 2018.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems(NIPS)*, 2012.

[18] Hongguang Zhang and Piotr Koniusz. Power normalizing second-order similarity network for few-shot learning. In *Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[19] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

[20] Pei Chen and David Suter. An analysis of linear subspace approaches for computer vision and pattern recognition. *International Journal of Computer Vision*, 68(1):83–106, 2006.