Visual Domain Bridge: A source-free domain adaptation for cross-domain few-shot learning

Moslem Yazdanpanah Parham Moradi
University of Kurdistan
{M.Yadanpanah, P.Moradi}@UOK.ac.ir

Abstract

Due to the covariate shift, deep neural networks performance always degrades when applied to novel domains. In order to mitigate this problem, domain adaptation techniques require samples from target data during the feature extraction training, which is not always applicable in real-world scenarios. Batch Normalization is a known component of computer vision models, aiming at reducing the training-time covariate shift. However, facing distribution shift results in an internal state mismatch inside the BatchNorm layers during the inference time. In favor of alleviating the induced mismatch, this paper proposes a source-free, lightweight and straightforward approach by introducing the "Visual Domain Bridge" concept reducing the BatchNorm's internal mismatch in the cross-domain settings. Compared to the other BatchNorm-based source-free domain adaptation techniques such as AdaBN and Prediction-BN, our method formed a new state-of-the-art cross-domain few-shot fine-tuning method neglecting extra augmentations; while improving the performance in near-domain settings too. The proposed method can integrate with other domain adaptation methods and enhance their performance requiring just a few lines of modification in the BatchNorm's implementation. Implementations are available in https://github.com/MosyMosy/VDB

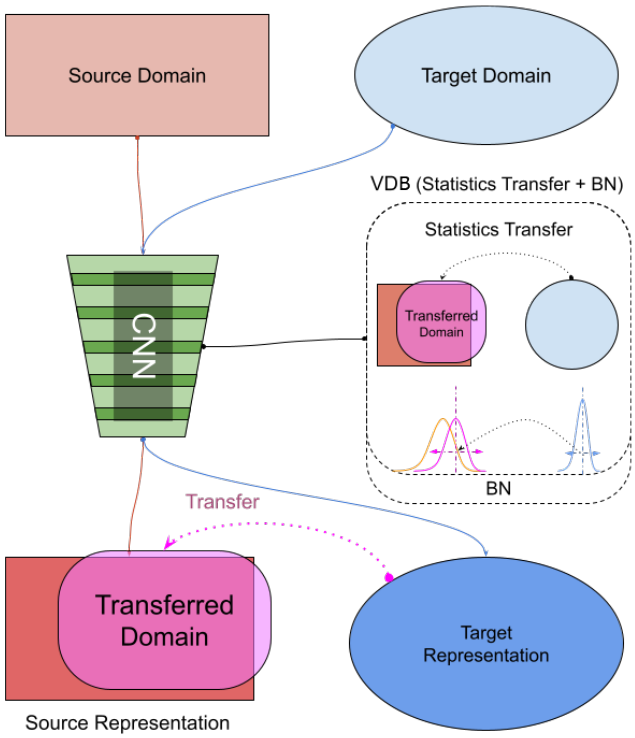

Figure 1. Illustration of the Visual Domain Bridge (VDB). Inside a CNN's Batch Normalization (BN) layers, the intermediate features get transferred to the source domain distribution when facing a target domain with domain shift.

1. Introduction

Although Convolutional Neural Network (CNN) [10] achieved outstanding results in different deep learning tasks, when facing domain shift, their performance falls drastically. The problem of applying a learned representation from a source domain to a target domain under the distribution shift gets even worse when the target domain is rarely annotated [4]. In order to mitigate this shortcoming, there is a large body of research as well as challenges, competitions, and benchmarks, conducted toward transfer learning across distant domains [31]. Domain adaption leveraging the statistical distribution of data is a well-studied technique in the literature [11]. However, most of these efforts focus on computationally expensive approaches benefiting target samples to make a robust representation under domain shift [20].

Batch Normalization (BN) [8] is a well-known component of deep CNN which speeds up model convergence during the training. Adding BN layers to deep learning models stabilizes the distribution of layer input features by mod-

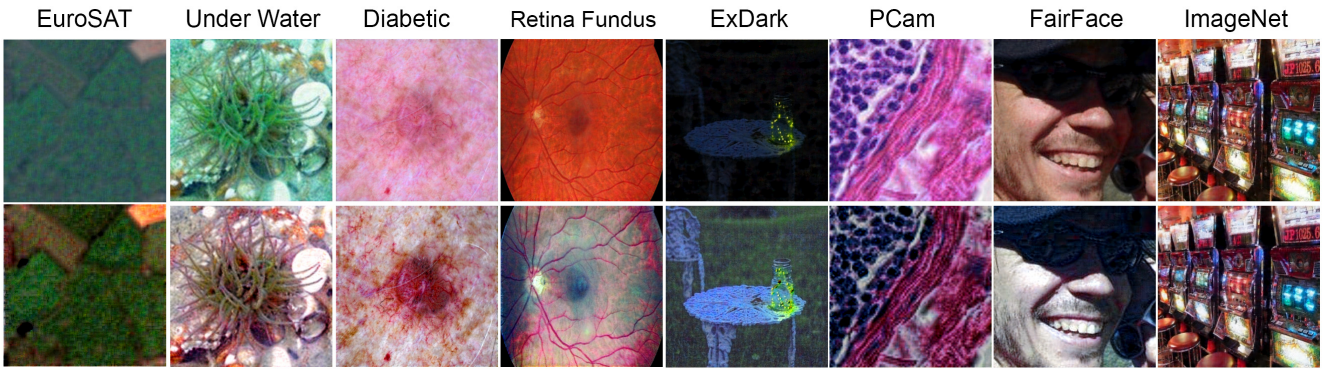

Figure 2. The output of a pre-trained autoencoder when fed with different datasets in inference time. The top row shows the output when all normalization layers are the Batch Normalization (BN). The bottom row depicts the output for the same inputs but, the BN layers are replaced with the VDB. Best viewed in color.

ulating their mean and variance. Inspired from "label information is usually stored in network's weight matrix and the statistics of the batch norm layer represents the domain-related knowledge." [13], there are attempts to utilize the BN layers toward filling the domain gap between the source and target domains. Utilizing the BN layers by the mean of domain adaptation as a tempting lightweight approach attracted a portion of the literature; Studies such as the AdaBN [13], Prediction-BN [17], TENT [24] and Core [30]. Despite the effectiveness of these proposed methods, they suffer from induced internal state mismatch. To alleviate this problem, they need steps of feature extractor update during the inference time. Both of these shortcomings have turned these methods into less real-world practical approaches.

Few-shot learning (FSL) as meta-testings followed by meta-learning steps refer to the problem of learning a task from just a few samples of the task's distribution. Despite its effectiveness and recent advances, the FSL underperform when there exists a significant shift between base and novel class domains [4]. This work proposes a source-free few-shot domain adaptation method by adapting the BN statistic as a lightweight interpretable approach, as depicted in Figure 1. Based on our knowledge, the Prediction-BN is the current state-of-the-art (SOTA) in source-free BN based methods, which does not utilize additional augmentations or backbone updating; the proposed method formed a new SOTA in cross-domain few-shot learning (CDFSL), outperforming the Prediction-BN on the ImageNet source CDFSL benchmark score, by more than two percents.

While most of the Domain Adaptation approaches aim at adapting a pre-trained model towards a target domain (with a possible distribution shift in input features); Inspired from [7], we propose a contradictory viewpoint by transferring the input's statistical elements to the training times accumulated statistics, inside BatchNorm Layers. This process accrues in test time without modifying the training procedure or the pre-trained backbone. From this, we attempt to decrease the internal state mismatch of the BNs, rooted in the shift between the source and target data (Figure 1).

The rest of the paper is structured as follows: First, the needed notations are represented, followed by the formal definition of the proposed method. Then we recall a brief history of the current related methods. After that, experiments setup and evaluation results are presented, followed by the analysis of the result. In the end, the paper is summed up with a conclusion.

2. Methodology

2.1. Preliminaries

For consistency with the literature, we adapt the notations from [25].

2.1.1 Domain

A domain $\mathcal{D}$ is composed of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\} \in \mathcal{X}$. Having a specific domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a task $\mathcal{T}$ consists of a feature space $\mathcal{Y}$ and a conditional probability distribution $P(Y|X)$; learnable in a supervised manner from the labeled data $\{x_i, y_i\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. Given two domains: a training dataset with adequate labeled data as the source domain $\mathcal{D}^s = \{\mathcal{X}^s, P(X)^s\}$, and a test dataset with insufficient labeled data as the target domain $\mathcal{D}^t = \{\mathcal{X}^t, P(X)^t\}$; with partially labeled part, $\mathcal{D}^{tl}$, and the unlabeled parts, $\mathcal{D}^{tu}$, form the entire target domain, that is, $\mathcal{D}^t = \mathcal{D}^{tl} \cup \mathcal{D}^{tu}$. Each domain comes with its task: The source task $\mathcal{T}^s = \{\mathcal{Y}^s, P(Y^s|X^s)\}$, and the target one is $\mathcal{T}^t = \{\mathcal{Y}^t, P(Y^t|X^t)\}$.

Distribution Shift (Domain shift) refers to situations where feature distribution differs from D^s to D^t: $P(X)^s \neq P(X)^t$

Near-Domain refers to situations where feature distribution are the same between D^s and D^t, but conditional probability distribution differs from $\mathcal{T}^s$ to $\mathcal{T}^t$: $P(X)^s = P(X)^t$ and $P(Y^s|X^s) \neq P(Y^t|X^t)$

Cross-Domain refers to situations where feature distribution differs from D^s to D^t while conditional probability distribution differs from $\mathcal{T}^s$ to $\mathcal{T}^t$ too: $P(X)^s \neq P(X)^t$ and $P(Y^s|X^s) \neq P(Y^t|X^t)$

2.1.2 Batch Normalization

Having a batch of labeled examples $\{(x_i^s, y_i^s)\}_{i=1}^N$ of size N from a source domain $\mathcal{D}^s$ where $x_i^s \in \mathcal{X}^s$ and $y_i^s \in \mathcal{Y}^s$, and Θ as a deep convolutional neural network consisting of L layers with weight matrices θ^l where l represents the layer index. If h represents the intermediate features of Θ for layer l, the Batch Normalization at layer l is computed for each channel and can be defined as:

$$\mathrm{BN}(h_c^s) = \gamma^s \times \frac{h_c^s - \mu_c^s}{\sqrt{{\sigma_c^2}^s + \epsilon}} + \beta^s \tag{1}$$

here, subscript c represents the channel index, γ^s and β^s are learnable affine parameters and μ_c^s and σ_c^s are statistical mean and variance of h_c^s respectively defined as:

$$\mu_c^s = \frac{1}{NHW} \sum\nolimits_{n,h,w} h_{nchw}^s \tag{2}$$

$$\sigma_c^s = \sqrt{\frac{1}{NHW} \sum\nolimits_{n,h,w} (h_{nchw}^s - \mu_c^s)^2}, \tag{3}$$

where H and W are the spatial dimensions of h_c^s.
The superscript s stood for the relation of features, statistical elements, and learned parameters with the source domain $\mathcal{D}^s$.

2.1.3 Batch Normalization under distribution shift

During the inference time, features $x_i^t \in \mathcal{X}^t$ from a target domain $\mathcal{D}^t$ with distribution shift to the source domain $\mathcal{D}^s$, are normalized using the training time domain's statistics and relevant fitted affine parameters as follow:

$$\mathrm{BN}(h_c^t) = \gamma^s \times \frac{h_c^t - \mu_c^s}{\sqrt{{\sigma_c^2}^s + \epsilon}} + \beta^s \tag{4}$$

This results in an inconsistency between the input futures h_c^t statistical elements and source domain captured ones inside the BN layers referred to as Internal State Mismatch [17]. We attempt to transfer the target features using the source and target domain's statistical elements to alleviate this internal state mismatch.

2.2. Proposed method

Assume there is $\mathcal{D}^s$ and $\mathcal{D}^t$ as cross-domains. While most of the Domain Adaptation approaches aim at adapting the $P(Y^s|X^s)$ toward the target domain $\mathcal{D}^t$, we propose a contradictory viewpoint by transferring the h_c^t's statistical mean and variance to the source domain $\mathcal{D}^s$ during the fine-tuning. From this, we attempt to decrease the internal state mismatch of the BNs, rooted in the shift between the $P(X)^s$ and $P(X)^t$ (Figure 1).

2.2.1 Visual Domain Bridge (VDB)

Having a target domain $\mathcal{D}^t$ with distribution shift according to the training time source domain $\mathcal{D}^s$, before normalizing the intermediate feature h_c^t we transfer it using the source and target domain statistical elements as follows:

$$h_c^{trans} = \frac{h_c^t - \hat{\mu_c}^t}{\sqrt{\hat{\sigma_c^2}^t + \epsilon}} \times {\sigma_c^2}^s + \mu_c^s \tag{5}$$

in order to reduce the imposed mismatch of new features' statistics (as reported in [17]), we employ a weighted average of the target and source statistic. The $\hat{\mu_c}^t$ and $\hat{\sigma_c^2}^t$ are calculated as follow:

$$\hat{\mu_c}^t = \alpha \mu_c^t + (1-\alpha)\, \mu_c^s \tag{6}$$

$$\hat{\sigma_c}^t = \alpha \sigma_c^t + (1-\alpha)\, \sigma_c^s \tag{7}$$

where α equals to the momentum factor as the BN's in training time. $\hat{\mu_c}^t$ and $\hat{\sigma_c}^t$ are the same as $\hat{\mu_c}^s$ and $\hat{\sigma_c}^s$, but on target feature h_{nchw}^s.

The batch normalization to the transferred intermediate feature h_c^{trans} is applied using the new emerging transferred domain $\mathcal{D}^{trans}$'s statistical elements:

$$\mathrm{BN}_{\mathrm{transfer}}(h_c^t) = \gamma^s \times \frac{h_c^{trans} - \hat{\mu_c}^{trans}}{\sqrt{\hat{\sigma_c^2}^{trans} + \epsilon}} + \beta^s \tag{8}$$

similarly, the $\hat{\mu_c}^{trans}$ and $\hat{\sigma_c^2}^{trans}$ are weighted average of statistic elements of the training-time source domain and the transferred features h_{nchw}^{trans}:

$$\hat{\mu_c}^{trans} = \alpha \mu_c^{trans} + (1-\alpha)\, \mu_c^s \tag{9}$$

$$\hat{\sigma_c}^{trans} = \alpha \sigma_c^{trans} + (1-\alpha)\, \sigma_c^s \tag{10}$$

Where

$$\mu_c^{trans} = \frac{1}{NHW} \sum\nolimits_{n,h,w} h_{nchw}^{trans} \tag{11}$$

$$\sigma_c^{trans} = \sqrt{\frac{1}{NHW} \sum\nolimits_{n,h,w} (h_{nchw}^{trans} - \mu_c^{trans})^2} \tag{12}$$

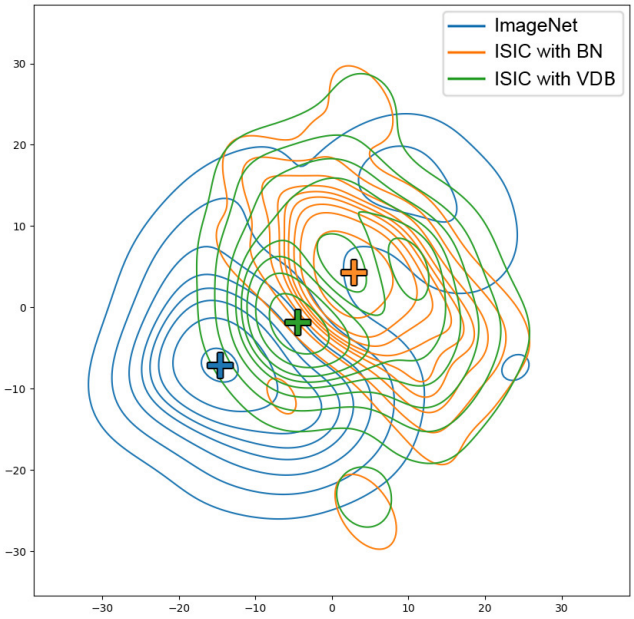

Figure 3. t-SNE plot of output logits of a ResNet18 pre-traine on ImageNet depicted as Kernel Density Estimations. Blue: with BN fed with the Imagenet, Orange: with BN fed with the ISIC, Green: with VDB fed with the ISIC. Hperparameters: learning-rate=200, iteration=1000, and perplexity=30. The + marks represent the density centers.

3. Related work

Despite the fame of meta-learning domain adaptation methods, Guo et al. [4] revealed that the pure fine-tuning exceeds current SOTA meta-learning performance when fronting a significant distribution shift. Problems with a significant domain gap between the source and target data are the main focus of recent research in FSL [4]. Also, progress in semi-supervised and self-supervised learnings methods resulted in advances for the CDFSL problems; They always need significant computational resources and multiple steps of adjustment. STARTUP [18] is a well-known technique in distant domain problems that utilizes a mixture of self-supervised and self-training elements for CDFSL. In contrast to the STARTUP, the proposed method is simpler and more lightweight by a large margin. Furthermore, a branch of domain adaptation methods is specially focused on employing batch normalization to bridge between domains [13]. AutoDIAL [3], and TransNorm [26] offered techniques to train the feature extractor parameters using a mixture of source and known target domains. However, the target domain is not always known during the training time. These methods are constrained to re-train the feature extractor, which is much heavier than just statistically transferring the input features. A recent work [6] offered a two-step proposal for transferring the input feature vector into a more Gaussian-shaped distribution followed by a transitive approach to adapting the feature vector to the test-time data. Despite its effectiveness, its transductive nature limits its application to new targets without re-adaptation, which is a limited approach compared to the proposed method. As a generalized source-free domain adaptation technique, [28] proposes a method to adapt to new domains while keeping the performance on the source domain. Source-free refers to methods independent of the training-time data during the adaptation phase. Focusing on the source-free methods as more practical domain adaptation approaches, the AdaBN [13] proposes adapting the BatchNorm's moments at test time based on the target's domain statistics, demanding access to the whole target samples. Although the AdaBN is not explicitly designed for the few-shot setting, it is known as an influential paper in the field of BatchNorm-based domain adaptation. Despite its effectiveness and simplicity, its performance decay when facing a distribution shift in test-time data. The AdaIN method [7] exploited the InstanceNorm [21] layers statistics by mean of style transfer. Although the style transfer is rarely related to the few-shot adaptation, the core idea is similar to the AdaBN and is a source of inspiration in this work. The Prediction-BN [17] proposes a similar method to AdaBN by replacing the batch-wise target statistics instead of the whole domain's statistics. The main motivation for this refinement is that the AdaBN imposes a large discrepancy between the BatchNorm's statistics and weight layers, which is alleviated by batch-wise statistics updates. Evaluations reveal that the Prediction-BN performance decrease through time and finally converges to the AdaBN's level. In order to adapt the backbone's wights to the imposed discrepancy from the BN's statistics replacement and Inspired by Prediction-BN, [24] added the BN's affine parameters adapting by entropy minimization named as TENT, which requires the backbone updating during the inference time. However, entropy minimization is not a good measure to keep the feature extractor's discrimination power. As TENT does not preserve the discriminative ability of the feature extractor, and the Prediction-BN causes a mismatch with the network state, [30] proposes a test-time calibration of the BN's statistics followed by an affine parameter adaption named as Core. Despite the effectiveness of the TENT and Core methods, both require backpropagation across the feature extractor during the inference time, which is not always applicable. This study investigated an alternative BN's statistic transfer in the CDFSL setting, exposing a slight internal state mismatch to the backbone but independent of updating the feature extractor. Recently the FeaturNorm [29] was proposed as an alternative for the BatchNorm layers during the training time, resulting in a more generalizable representation under distribution-shift. As this method neglects the learnable affine parameters of the BN layers, it needs the

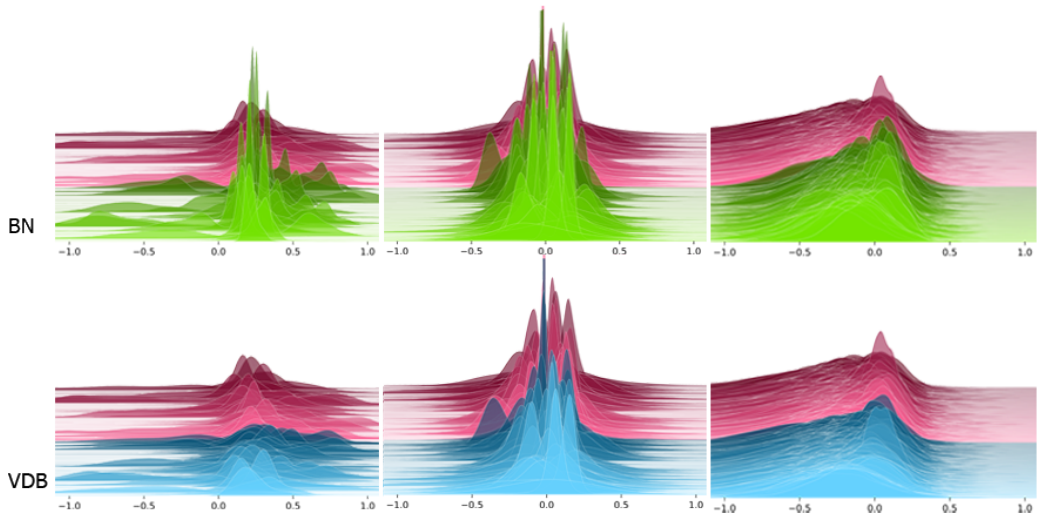

Figure 4. Output distribution of selected Normalization layers from a ResNet18, pre-trained on ImageNet (Best viewed in color). Red: ImageNet on BN layers, Blue: EuroSAT on BN Layers, and Green: EuroSat on VDB layers. Left: layer 0, Center: layer 7, and Right: layer 18 which are selected based on a uniform probability. When facing distribution shift, the model with VDB results in more picked and centered distributions in per-layer aggregated feature values. Also it as apparent that VDB maintains the distribution shape when fed with training time domain. We witnessed the same pattern across all layers.

backbone to be trained from scratch and is not comparable with our source-free and simple method. However, we have evaluated the combination of the FeatureNorm with our proposed method to demonstrate the ability of our method to combine with other straightforward approaches. Compared to the AdaBN, Prediction-BN, TENT, and Core, our method transfers the statistical elements of the input feature vector. It does not exert any modification or update to the backbone, making it a suitable lifelong and source-free domain adaptation method.

4. Evaluation

4.1. Why VDB work?

Visualization is a convenient and interpretable approach to understanding method effects. Here, the VDB is visualized to investigate the effect of the proposed method on the feature representation of an encoder component of a convolutional autoencoder architecture. Then we present the representation and normalization layers' output distributions of a pre-trained model when facing the training-time source domain and an arbitrary target domain both for BN and VDB as normalization layers.

Experiment Setup: We utilized a SegNet [1] based convolutional autoencoder from [14] and replaced the encoder part's BN layers with the VDB, just before loading the pre-trained state dictionary. As the state dictionary is pre-trained on the ImageNet like datasets, a visual transfer of any input to the real-world day-time color schema is expected. We feed batches of samples from challenging datasets (EuroSAT [5], Under Water [12], Diabetic Retinopathy [27], HRF [2], ExDark [15], PCam [22], FairFace [9], and the ImageNet as blank domain) first with BN for all of the normalization layers and then with VDB replaced in encoder part (momentum set to 1). The result is illustrated in Figure 2. An empirical evaluation of the BN and VDB output's distribution is conducted using a publicly available pre-trained ResNet18 fed with batches of samples from the ImageNet as the source domain and the EuroSAT as the target domain. The output values of normalization layers are channel-wise aggregated as presented in Figure 4. In Figure 3, the representation gap induced from the distribution gap is considered. We plot the calculated t-SNE[1] from logits of a pre-trained ResNet18, equipped with the BN layers fed with samples from the source domain and the ISIC. Then, the Kernel Density Estimation of the t-SNE plot is compared with the same model where the BN layers are replaced with the VDB and fed with the ISIC domain.

Results: From Figure 2, it is evident that the autoencoder with VDB visually transfers any arbitrary target domains to the source ImageNet like domain. From the right side of the figure, the sample from ImageNet remains unchanged, whereas the samples from other domains visually get transferred toward the source domain, resulting in a lower distribution gap in CNN's layers inputs. The effect of VDB is demonstrated as more picked distributions at the normaliza-

[1] t-distributed stochastic neighbor embedding

	5 Ways # Shots	CropDisease	EuroSAT	ISIC	ChestX
pure tuning	1	68.46 ± 0.87	59.18 ± 0.85	33.11 ± 0.60	22.54 ± 0.42
AdaBN		68.19 ± 0.85	57.46 ± 0.80	34.72 ± 0.65	22.32 ± 0.41
Prediction-BN		70.57 ± 0.84	61.60 ± 0.83	35.52 ± 0.65	22.60 ± 0.42
VDB (our)		71.98 ± 0.82	63.60 ± 0.87	35.32 ± 0.65	22.99 ± 0.44
STARTUP		74.56 ± 0.85	64.00 ± 0.88	35.12 ± 0.64	22.93 ± 0.43
STARTUP + VDB		77.43 ± 0.81	63.61 ± 0.87	35.54 ± 0.64	23.20 ± 0.43
FN		70.93 ± 0.87	62.75 ± 0.88	32.77 ± 0.60	22.37 ± 0.41
FN + VDB		74.75 ± 0.79	65.60 ± 0.89	34.94 ± 0.63	22.54 ± 0.43
MAML*	5	78.05 ± 0.68	71.70 ± 0.72	40.13 ± 0.58	23.48 ± 0.96
ProtoNet*		79.72 ± 0.67	73.29 ± 0.71	39.57 ± 0.57	24.05 ± 1.01
pure tuning		89.86 ± 0.50	79.99 ± 0.64	45.53 ± 0.59	26.66 ± 0.42
AdaBN		90.12 ± 0.50	80.21 ± 0.60	48.88 ± 0.62	25.66 ± 0.42
Prediction-BN		89.80 ± 0.51	81.71 ± 0.57	50.42 ± 0.65	25.61 ± 0.43
VDB (our)		90.77 ± 0.49	82.06 ± 0.63	48.72 ± 0.65	26.62 ± 0.45
STARTUP		92.86 ± 0.43	82.51 ± 0.62	48.54 ± 0.63	27.17 ± 0.44
STARTUP + VDB		93.13 ± 0.45	82.11 ± 0.64	49.72 ± 0.65	27.48 ± 0.45
FN		91.32 ± 0.46	80.75 ± 0.63	44.80 ± 0.57	26.32 ± 0.43
FN + VDB		92.20 ± 0.45	82.68 ± 0.61	47.79 ± 0.60	26.56 ± 0.46
MAML*	20	89.75 ± 0.42	81.95 ± 0.55	52.36 ± 0.57	27.53 ± 0.43
ProtoNet*		88.15 ± 0.51	82.27 ± 0.57	49.50 ± 0.55	28.21 ± 1.15
pure tuning		96.01 ± 0.28	88.02 ± 0.47	55.64 ± 0.57	31.98 ± 0.44
AdaBN		96.26 ± 0.28	88.94 ± 0.44	58.98 ± 0.58	31.11 ± 0.46
Prediction-BN		95.92 ± 0.28	89.82 ± 0.41	60.43 ± 0.61	30.91 ± 0.45
VDB (our)		96.36 ± 0.27	89.42 ± 0.45	59.09 ± 0.59	31.87 ± 0.44
STARTUP		97.43 ± 0.23	89.63 ± 0.43	59.98 ± 0.59	33.54 ± 0.46
STARTUP + VDB		97.42 ± 0.22	89.65 ± 0.45	60.37 ± 0.57	33.49 ± 0.49
FN		96.68 ± 0.25	87.88 ± 0.45	56.43 ± 0.56	32.36 ± 0.46
FN + VDB		97.09 ± 0.25	89.70 ± 0.43	59.51 ± 0.57	32.31 ± 0.46
ProtoNet*	50	90.81 ± 0.43	80.48 ± 0.57	51.99 ± 0.52	29.32 ± 1.12
pure tuning		97.59 ± 0.22	91.04 ± 0.37	61.38 ± 0.57	35.28 ± 0.48
AdaBN		97.90 ± 0.19	92.08 ± 0.36	63.61 ± 0.59	34.21 ± 0.45
Prediction-BN		97.61 ± 0.20	92.62 ± 0.33	64.34 ± 0.55	34.14 ± 0.46
VDB (our)		97.89 ± 0.19	92.24 ± 0.35	64.02 ± 0.58	35.55 ± 0.45
STARTUP		98.53 ± 0.16	92.59 ± 0.33	65.90 ± 0.56	37.67 ± 0.47
STARTUP + VDB		98.48 ± 0.17	92.46 ± 0.35	65.21 ± 0.55	38.17 ± 0.46
FN		98.09 ± 0.19	91.01 ± 0.38	62.64 ± 0.57	36.32 ± 0.47
FN + VDB		98.35 ± 0.17	92.29 ± 0.34	65.81 ± 0.56	36.90 ± 0.48
pure tuning	avg	87.98 ± 0.39	79.56 ± 0.43	48.91 ± 0.43	29.11 ± 0.37
AdaBN		88.12 ± 0.38	79.67 ± 0.42	51.55 ± 0.44	28.33 ± 0.37
Prediction-BN		88.47 ± 0.38	81.44 ± 0.41	**52.68 ± 0.44**	28.32 ± 0.38
VDB (our)		**89.25 ± 0.38**	**81.83 ± 0.43**	51.79 ± 0.44	**29.26 ± 0.38**
STARTUP		90.85 ± 0.37	**82.18 ± 0.43**	52.39 ± 0.44	30.33 ± 0.38
STARTUP + VDB		**91.62 ± 0.36**	81.96 ± 0.43	**52.71 ± 0.44**	**30.58 ± 0.38**
FN		89.26 ± 0.38	80.60 ± 0.43	49.16 ± 0.43	29.35 ± 0.38
FN + VDB		**90.60 ± 0.36**	**82.57 ± 0.43**	**52.01 ± 0.44**	**29.58 ± 0.38**

Table 1. The CDFSL classification accuracy results of a ResNet10 backbone pre-trained on miniImageNet for 5 ways and # number of shots. FN refers to the FeatureNorm method. avg: Average over all shots. * refers to results from [4].

	5 Ways # Shots	CropDisease	EuroSAT	ISIC	ChestX
pure tuning	1	71.71 ± 0.87	64.58 ± 0.82	30.30 ± 0.52	22.16 ± 0.40
AdaBN		67.00 ± 0.84	55.58 ± 0.86	29.43 ± 0.51	21.75 ± 0.38
Prediction-BN		69.84 ± 0.77	59.72 ± 0.86	31.65 ± 0.54	21.55 ± 0.39
VDB (our)		75.46 ± 0.76	67.76 ± 0.83	33.22 ± 0.58	22.28 ± 0.41
FN		76.98 ± 0.83	66.45 ± 0.78	30.09 ± 0.57	23.14 ± 0.42
FN + VDB		79.68 ± 0.74	69.67 ± 0.80	32.96 ± 0.57	22.64 ± 0.41
pure tuning	5	91.20 ± 0.46	84.88 ± 0.57	44.01 ± 0.57	25.23 ± 0.43
AdaBN		91.09 ± 0.47	77.73 ± 0.67	43.26 ± 0.55	23.94 ± 0.40
Prediction-BN		91.30 ± 0.43	80.10 ± 0.63	44.11 ± 0.52	24.00 ± 0.41
VDB (our)		93.11 ± 0.42	85.29 ± 0.52	47.48 ± 0.61	25.25 ± 0.42
FN		93.54 ± 0.42	86.10 ± 0.51	44.48 ± 0.61	26.15 ± 0.41
FN + VDB		94.63 ± 0.37	87.31 ± 0.50	47.48 ± 0.59	25.55 ± 0.43
pure tuning	20	96.52 ± 0.27	91.51 ± 0.37	55.66 ± 0.57	29.19 ± 0.41
AdaBN		96.84 ± 0.25	86.69 ± 0.53	55.33 ± 0.53	27.95 ± 0.41
Prediction-BN		96.91 ± 0.24	87.61 ± 0.48	55.35 ± 0.52	27.43 ± 0.42
VDB (our)		97.61 ± 0.21	91.93 ± 0.37	58.89 ± 0.59	29.49 ± 0.42
FN		97.62 ± 0.23	92.28 ± 0.33	56.61 ± 0.59	30.14 ± 0.44
FN + VDB		98.20 ± 0.19	92.85 ± 0.34	59.65 ± 0.58	30.28 ± 0.45
pure tuning	50	98.05 ± 0.20	93.34 ± 0.28	61.42 ± 0.56	31.80 ± 0.47
AdaBN		97.97 ± 0.19	89.63 ± 0.43	59.83 ± 0.55	29.75 ± 0.45
Prediction-BN		97.89 ± 0.22	90.55 ± 0.39	60.25 ± 0.53	29.12 ± 0.45
VDB (our)		98.40 ± 0.16	93.95 ± 0.30	64.23 ± 0.58	32.37 ± 0.47
FN		98.56 ± 0.17	94.30 ± 0.27	63.72 ± 0.57	33.60 ± 0.46
FN + VDB		98.83 ± 0.13	94.79 ± 0.27	65.27 ± 0.56	33.24 ± 0.45
pure tuning	Average	89.37 ± 0.38	83.58 ± 0.40	47.85 ± 0.42	27.10 ± 0.37
AdaBN		88.23 ± 0.37	77.40 ± 0.45	46.96 ± 0.42	25.85 ± 0.36
Prediction-BN		88.99 ± 0.36	79.49 ± 0.43	47.84 ± 0.41	25.53 ± 0.36
VDB (our)		**91.15 ± 0.35**	**84.73 ± 0.40**	**50.96 ± 0.43**	**27.35 ± 0.37**
FN		91.68 ± 0.36	84.78 ± 0.39	48.73 ± 0.43	**28.26 ± 0.37**
FN + VDB		**92.83 ± 0.34**	**86.15 ± 0.39**	**51.34 ± 0.43**	27.93 ± 0.37

Table 2. The CDFSL results for a ResNet18 backbone pre-trained on ImageNet, evaluated over 600 episodes. pure tuning, linear classifier fine-tuned on labeled target samples. AdaBN is fine-tuned the same as the pure tuning except adapt the BN layers before fine-tuning, Prediction-BN and VDB similar to the AdaBN except for the BN adaptation occurs during the fine-tuning. Results are for 5-Way and # Shots classification accuracy. avg: Average over all shots.

	1-shot	5-shot	20-shot
pure tuning	54.56 ± 0.84	76.18 ± 0.69	84.53 ± 0.52
VDB (our)	**56.08 ± 0.86**	**76.61 ± 0.72**	**84.59 ± 0.55**

Table 3. Near-domain few-shot evaluation on pure tuning and VDB. Models are pre-trained on miniImageNet and evaluated on novel classes of ImageNet over 600 episodes.

tion layers outputs as illustrated in Figure 4. From this figure, we can see VDB maintains the distribution shape when fed with training time domain while the model with VDB results in more picked and centered distributions in per-layer aggregated feature value when facing distribution shift (we witnessed the same pattern for all the layers). From Figure 3 it is observant that the model with VDB reduces the representation gap on data with distribution-shift. Com-

pared to the model with BN (orange), the VDB equipped green shape is expanded toward the source's shape (blue), and the center of the density is moved to the source domain's center of density. As previously explored in [19] distribution shift will appear as distance in representation, and reducing the representation distance will directly result in a performance upgrade. We repeated the plot with different hyperparameters resulted in similar patterns.

4.2. VDB in CDFSL

Experiment Setup: The FSL evaluation on the CDFSL benchmark is established as introduced in [4]. The miniImageNet [23] is employed as the base training-time source dataset as well as the more extensive ImageNet dataset. The target datasets are comprised of four datasets, each from a shifted distribution relative to the source domain (miniImageNet and ImageNet). The target datasets are composed of EuroSAT (satellite imagery to determine land usage), CropDiseases (plant images to identify botanical diseases), ChestX (chest X-rays to detect pathology), and ISIC2018 (images of skin abrasions to detect melanoma) as described in [4]. Methods such as STARTUP and AdaBN benefit randomly sampled 20% of unlabelled images from the target datasets to use in the representation learning stage, consistent with the setup of [18], remaining samples are used during the fine-tuning. Identical to [4], we conduct experiments in an FSL classification setting while the support set is comprised of 5 classes with k samples per class (5-way k-shot), where k $\in$ {1, 5, 20, 50} and the overall CDFSL score is an average of accuracies across all target datasets for k $\in$ {5, 20, 50}. Evaluation of pre-trained models brings off over 600 episodes. 95% confidence intervals with mean accuracy are reported. Pure tuning stands for our baseline, is the method without any adaptation rather than the fine-tuning, which utilizes a pre-trained model regularly trained on the source domain. The CDFSL evaluations for ResNet10 pre-trained on the miniImageNet and ResNet18 pre-trained on ImageNet, are reported in Tables 1 and 2, respectively.

Results: When facing Cross-Domain targets, a ResNet10 architecture pre-trained on miniImageNet is evaluated based on the CDFSL benchmark's setting as presented in Table 1. From the reported values, we can see the VDB outperformed the last SOTA method, the Prediction-BN with a subtle margin. Also the VDB is able to improve the STARTUP method as a fancy domain adaptation method with heavy training process utilizing target samples and benefits from semi-supervised, and self-supervised losses beside the supervised one. Surprisingly, it is observant that the VDB rather than enhancing the FeaturNorm method, FeaturNorm + VDB perform near the STARTUP, which is a much heavier method. The CDFSL scores for Table 1 methods is: pure tuning (66.58 $\pm$ 0.67), AdaBN (67.33 $\pm$ 0.66), Prediction-BN (67.78 $\pm$ 0.66), **VDB (67.88 $\pm$ 0.82)** , STARTUP (68.86 $\pm$ 0.65), **STARTUP + VDB (68.97 $\pm$ 0.81)**, FN (67.05 $\pm$ 0.66), and FN + VDB (68.43 $\pm$ 0.81). When it comes to the common and more practical source domain in real-world scenarios, the ImageNet, we utilized a publicly available pre-trained ResNet18 and evaluated it in the CDFSL settings and on the target datasets. As it is evident from Table 2, Although the AdaBN and Prediction-BN, both underachieved the pure tuning, The VDB improved the pure tuning's performance by more than one percent in the CDFSL score. A gain roughly in free, capable of integrating with other domain adaptation methods. The CDFSL scores for Table 2 methods is: pure tuning (66.90 $\pm$ 0.64), AdaBN (65.00 $\pm$ 0.66), Prediction-BN (65.39 $\pm$ 0.65), **VDB (68.17 $\pm$ 0.78)**, FN (68.09 $\pm$ 0.63), and FN + **VDB (69.01 $\pm$ 0.77)**.

4.3. VDB in near-domain

Catastrophic forgetting [16] as a caveat of domain adaptation techniques in a way that a comprehension model tends to perform poorly in the source domain is a well-known phenomenon among domain adaptation methods. In order to examine the effect of VDB on forgetting the source domain, we evaluated a pre-trained ResNet10 on miniImageNet in Near-Domain setting in which $P(X)^s = P(X)^t$ and $P(Y^s|X^s) \neq P(Y^t|X^t)$, with the same setting as the CDFSL benchmark but on novel classes of ImageNet datasets as target domain.
From Table 3, although it was not expected, the VDB maintains the in-distribution performance even with a slight superiority compared to the same pre-trained model without any modifications.

5. Conclusion

This work proposes a source-free domain-adaptation method, the VDB, in the cross-domain few-shot learning setting. A modification to the BN layer to be substituted with it across a CNN in fine-tuning time. Empirical evaluations reveal the effectiveness of the VDB against the domain gap through the reduction in the representation gap. Also, it performed as a new SOTA approach among the BN-based source-free domain-adaptation methods on a sever few-shot benchmark, the CDFSL. The VDB improved the performance on miniImageNet as well as ImageNet as a more general source domain wherein previous SOTA methods underperform the pure fine-tuning.

Acknowledgments

All the experiments recorded in Tables 3, 1, and 2 are conducted using Google Colab, the free version.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 5

[2] Attila Budai, Rüdiger Bock, Andreas Maier, Joachim Hornegger, and Georg Michelson. Robust vessel segmentation in fundus images. *International journal of biomedical imaging*, 2013, 2013. 5

[3] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. Autodial: Automatic domain alignment layers. In *2017 IEEE international conference on computer vision (ICCV)*, pages 5077–5085. IEEE, 2017. 4

[4] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *European Conference on Computer Vision*, pages 124–141. Springer, 2020. 1, 2, 4, 6, 8

[5] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 5

[6] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. In *International Conference on Artificial Neural Networks*, pages 487–499. Springer, 2021. 4

[7] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 2, 4

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 1

[9] Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1548–1558, 2021. 5

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1

[11] Chuanzi Li, Jining Feng, Li Hu, Junhong Li, and Haibin Ma. Review of image classification method based on deep transfer learning. In *2020 16th International Conference on Computational Intelligence and Security (CIS)*, pages 104–108. IEEE, 2020. 1

[12] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. An underwater image enhancement benchmark dataset and beyond. *IEEE Transactions on Image Processing*, 29:4376–4389, 2020. 5

[13] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018. 2, 4

[14] Yang Liu. Convolutional autoencoder with setnet in pytorch. https://github.com/foamliu/Autoencoder, 2018. 5

[15] Yuen Peng Loh and Chee Seng Chan. Getting to know low-light images with the exclusively dark dataset. *Computer Vision and Image Understanding*, 178:30–42, 2019. 5

[16] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 8

[17] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020. 2, 3, 4

[18] Cheng Perng Phoo and Bharath Hariharan. Self-training for few-shot transfer across extreme task differences, 2021. 4, 8

[19] Karin Stacke, Gabriel Eilertsen, Jonas Unger, and Claes Lundström. Measuring domain shift for deep learning in histopathology. *IEEE journal of biomedical and health informatics*, 25(2):325–336, 2020. 8

[20] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *arXiv preprint arXiv:2007.00644*, 2020. 1

[21] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[22] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. June 2018. 5

[23] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638, 2016. 8

[24] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 2, 4

[25] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2

[26] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable normalization: Towards improving transferability of deep neural networks. 2019. 4

[27] Zhiguang Wang and Jianbo Yang. Diabetic retinopathy detection via deep convolutional networks for discriminative localization and visual explanation. In *Workshops at the thirty-second AAAI conference on artificial intelligence*, 2018. 5

[28] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021. 4

[29] Moslem Yazdanpanah, Aamer Abdul Rahman, Muawiz Chaudhary, Christian Desrosiers, Mohammad Havaei, Eugene Belilovsky, and Samira Ebrahimi Kahou. Revisiting learnable affines for batch norm in few-shot transfer learning.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision [Manuscript submitted for publication]*, 2022. 4

[30] Fuming You, Jingjing Li, and Zhou Zhao. Test-time batch statistics calibration for covariate shift. *arXiv preprint arXiv:2110.04065*, 2021. 2, 4

[31] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. The visual task adaptation benchmark. 2019. 1