Channel Importance Matters in Few-Shot Image Classification

Xu Luo[1], Jing Xu[2], Zenglin Xu[2]
[1]University of Electronic Science and Technology of China [2]Harbin Institute of Technology Shenzhen
Frank.Luox@outlook.com {xujing.may, zenglin}@gmail.com

Abstract

Few-Shot Learning (FSL) requires vision models to quickly adapt to brand-new classification tasks with a shift in task distribution. Understanding the difficulties posed by this task distribution shift is central to FSL. In this paper, we show that a simple channel-wise feature transformation may be the key to unraveling this secret from a channel perspective. When facing novel few-shot tasks in the test-time datasets, this transformation can greatly improve the generalization ability of learned image representations, while being agnostic to the choice of training algorithms and datasets. Through an in-depth analysis of this transformation, we find that the difficulty of representation transfer in FSL stems from the severe channel bias problem of image representations: channels may have different importance in different tasks, while convolutional neural networks are likely to be insensitive, or respond incorrectly to such a shift. This points out a core problem of modern vision systems and needs further attention in the future.

1. Introduction

Few-Shot Learning (FSL) challenges current vision models on the ability to quickly adapt to novel few-shot tasks that are different from those in training. This *task distribution shift* means that categories, domains of images or granularity of categories in new tasks deviate from those in the training tasks. Recent studies of few-shot image classification have highlighted the importance of the quality of learned image representations [4, 5, 11, 12, 15], and also showed that representations learned by neural networks do not generalize well to novel few-shot classification tasks when there is task distribution shift [1, 3, 5]. Thus it is crucial to understand how task distribution shift affects the generalization ability of image representations in FSL.

In this paper, we show that a channel-wise feature transformation may be the key to this question. This transformation function is applied to *pre-trained* image representations channel-wisely only at test-time on the fly. Empirical studies show that the transformation function harms in-distribution FSL performance, but can consistently and largely improve predictions for out-of-distribution FSL tasks, being agnostic to the choice of algorithms, training and test-time datasets.

Through an in-depth analysis of the presented simple transformation, we reveal that the task distribution shift leads to a *channel bias* problem inside the representations (features) learned by convolutional neural networks. Specifically, in the layer after global pooling, different channels in the learned feature seek for different patterns (as verified in [2, 17]) during training, and the channels are weighted (in a biased way) based on their importance to the training task. However, when applied to novel few-shot classification tasks, the learned image features usually do not change much or have inappropriately changed without adapting to categories in novel tasks. This bias towards training tasks may result in imprecise attention to image features in novel tasks. We show the extent of this channel bias on different test-time datasets and show how the simple transformation alleviates this problem.

2. A Channel-wise Feature Transformation

2.1. Problem Setup

In few-shot image classification, a training set $\mathcal{D}^{train}$ is used at first to train a neural network parameterized by θ, which is evaluated on a series of few-shot classification tasks constructed from the test-time dataset $\mathcal{D}^{test}$. Importantly, there should be task distribution shift between $\mathcal{D}^{train}$ and $\mathcal{D}^{test}$. Each evaluated N-way K-shot few-shot classification task τ has a support set $\mathcal{S}_\tau$ consisting of N classes of images sampled from $\mathcal{D}^{test}$, each of which contains K images. $\mathcal{S}_\tau$ is used to construct a classifier $p_\theta(\cdot|x, \mathcal{S}_\tau)$ which is further evaluated on the unlabeled query set $\mathcal{Q}_\tau$ sampled from the same N classes.

2.2. Universal Performance Gains with a Simple Feature Transformation

We consider a simple channel-wise feature transformation as follows

Table 1. **5-way 5-shot performance gains of the simple feature transformation**. The black values indicate the original accuracy, and the red values indicate the increase. CE: conventional training with cross-entropy loss.

	TrainData	*mini*-train							ImageNet			iNaturalist	
TestData	Algorithm	PN [14]	PN	CE	MetaB [3]	MetaOpt [8]	CE	S2M2 [9]	PN	CE	MoCo-v2 [7]	CE	Average
	Architecture	Conv-4	Res-12	Res-12	Res-12	Res-12	SE-Res50	WRN	Res-50	Res-50	Res-50	Res-50	
MetaDataset [16]	*mini*-test	66.6+1.2	73.5+2.2	75.9+1.6	74.7+2.6	74.8+0.5	76.2+0.2	82.5+1.2	82.2-1.6	89.1-0.5	93.7+2.2	69.9+2.2	78.1+1.1
	CUB	52.0+2.8	57.0+3.0	59.6+2.3	60.1+2.6	60.3+1.7	59.9+2.2	68.5+2.8	65.3+2.5	78.2+0.4	70.0+6.8	94.7+0.0	66.0+2.5
	Textures	50.9+2.3	57.1+4.2	63.1+2.4	61.2+3.7	60.2+1.8	63.5+0.6	69.3+2.9	61.9+2.4	71.6+0.8	82.8+0.9	63.2+2.3	64.1+2.0
	Traffic Signs	52.6+2.1	64.8+2.2	65.6+1.4	67.3+1.5	67.1+4.9	62.2+2.9	69.6+3.1	64.0+2.2	67.2+3.5	68.4+8.8	60.5+4.0	64.4+3.3
	Aircraft	32.1+0.9	31.3+1.6	34.7+1.9	34.7+2.3	35.6+2.4	38.2+2.0	40.5+4.7	38.4+1.7	46.6+2.5	34.5+8.8	42.1+2.5	34.0+2.9
	Omniglot	61.0+10.0	77.6+7.8	86.9+3.7	81.6+7.9	78.0+9.9	89.9+2.3	85.9+7.4	76.4+2.9	88.6+5.3	74.5+15.8	83.8+9.0	80.4+7.5
	VGG Flower	71.0+3.1	71.1+5.5	79.2+3.8	78.3+4.5	78.4+3.1	83.0+1.7	87.8+2.5	81.4+2.6	89.3+1.7	86.2+6.3	91.9+1.1	81.6+3.3
	MSCOCO	52.0+1.2	58.2+1.1	59.0+0.7	58.0+1.6	58.4+0.1	57.1+0.5	63.5+0.1	61.3-0.5	64.3-0.4	71.4+1.4	50.4+1.9	59.4+0.7
	Quick Draw	49.7+6.5	60.2+5.4	67.5+6.5	61.9+9.0	61.0+6.2	69.8+2.8	66.4+8.2	59.8+6.9	70.2+3.0	63.7+8.3	60.8+6.2	62.8+6.3
	Fungi	48.5+1.5	49.0+3.7	52.2+3.3	51.5+4.0	54.6+1.9	55.2+0.5	61.6+3.8	58.5+1.3	65.1+1.1	60.2+9.2	70.0+1.8	56.9+2.9
BSCD-FSL [6]	Plant Disease	66.6+7.8	73.3+7.9	80.0+5.1	75.6+7.6	78.6+4.5	83.1+3.2	86.4+3.5	72.5+8.0	84.1+3.3	87.1+4.7	85.6+4.1	79.4+5.4
	ISIC	38.5+1.6	36.8+2.9	40.4+1.0	38.8+1.7	39.5+2.3	37.7+3.9	40.5+5.5	39.5+4.0	37.8+3.6	43.2+2.8	39.0+4.3	39.2+3.1
	EuroSAT	63.0+4.5	67.3+5.5	75.7+2.9	71.9+4.5	72.8+5.8	75.7+1.6	81.2+2.9	72.5+6.1	78.4+2.2	83.5+2.7	73.5+3.7	74.1+3.9
	ChestX	22.9+0.2	23.0+0.5	24.1+0.3	23.5+0.5	24.5+0.4	23.6+0.2	24.2+0.9	23.2+0.3	24.2+0.8	25.4+0.9	23.9+0.1	23.9+0.5
DomainNet [10]	Real	67.0+1.8	72.2+3.1	76.3+1.6	75.0+2.6	75.8+1.1	76.7+0.5	81.7+1.9	80.5+0.4	87.1-0.1	88.8+2.1	72.9+1.7	77.6+1.5
	Sketch	42.6+2.9	45.3+5.0	51.1+2.6	50.2+3.4	50.6+2.0	50.9+2.4	56.8+4.1	53.1+1.5	63.2+2.5	63.9+5.8	51.9+1.4	52.7+3.1
	Infograph	33.1+2.8	34.7+3.7	35.3+2.8	35.0+4.0	38.3+1.1	38.2+2.5	39.2+3.7	39.7+2.7	42.3+4.2	41.6+7.1	38.5+2.9	37.8+3.4
	Painting	49.0+1.7	52.5+3.3	56.1+1.4	55.1+2.5	56.2+0.7	59.3+0.8	64.2+1.8	61.8-0.2	69.6+0.5	76.5+3.0	56.4+1.9	59.7+1.6
	Clipart	47.5+3.6	49.7+4.8	55.5+3.1	54.9+4.3	56.4+2.6	60.4+2.3	63.0+4.3	60.9+1.8	72.7+1.5	67.4+7.0	58.4+2.2	58.8+3.4

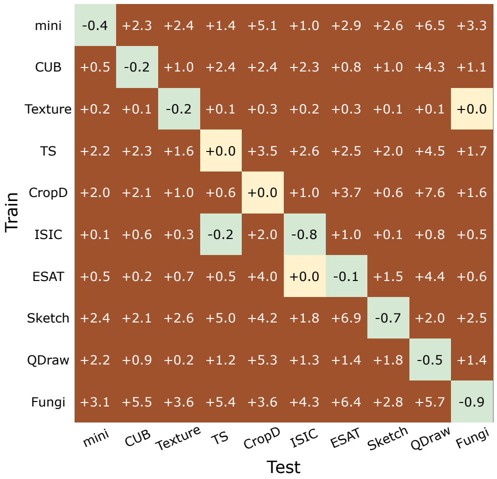

Figure 1. In-distribution (diagonal) and out-of-distribution (off-diagonal) performance gains of the simple channel-wise transformation on representations trained with CE. When the test-time dataset equals the training dataset (diagonal), the categories of images remain the same but test-time images are unseen during training (as in conventional classification).

$$\phi(x) = \begin{cases} \frac{1}{ln^k(\frac{1}{|x|}+1)}, & x \neq 0 \\ 0, & x = 0 \end{cases} \tag{1}$$

where $k > 0$ is a hyperparameter. We append this transformation function on the top of any *pre-trained* feature extractor $f_\theta(\cdot)$ channel-wisely only at test-time, which essentially applies a non-linear transformation to the *learned representations* just after the global pooling layer.

In Table 1, we show the performance gains with this transformation on 5-way 5-shot FSL tasks. We test the transformation on representations trained with various datasets, algorithms and backbone networks. We fix $k = 1.3$ (We show how performance varies with different choices of k in the appendix). Details can be found in the appendix. As seen, the simple feature transformation substantially improves few-shot learning across various algorithms, datasets and architectural choices. The only exception happens when the test-time task distribution is very similar to a subset of training distribution: training the supervised models on ImageNet and testing on *mini*ImageNet, MSCOCO, Real or Painting, or training on iNaturalist and testing on CUB. To further verify this, we train a CE model on each of ten datasets and test on 5-way 5-shot tasks sampled from each dataset. The results shown in Figure 1 give evidence that the transformation is beneficial only to few-shot classification with task distribution shift—the performance is improved only when test-time task distribution deviates from training, and this distribution shift includes domain shift (e.g., from Sketch to QuickDraw), category shift (e.g., from Plant Disease to Fungi) and granularity shift (e.g., from iNaturalist to Plant Disease in Table 1).

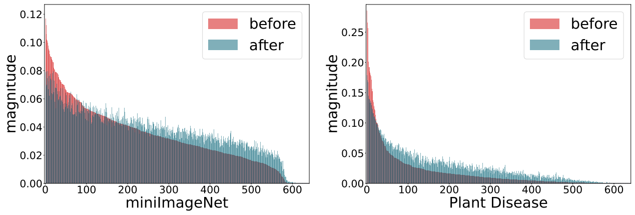

Figure 2. **Mean magnitudes of feature channels before and after the simple transformation.** The feature extractor is trained using PN on the training set of *mini*ImageNet.

3. The Channel Bias Problem

In this section, we analyze the presented simple transformation and point out alleviating the channel bias problem when facing task distribution shift is the key to its success. It can be first noticed that

$$\phi'(x) > 0, \lim_{x \to 0^+} \phi'(x) = +\infty,$$
$$\exists t > 0, \quad s.t. \quad \forall x \in (0, t), \phi''(x) < 0, \tag{2}$$

Table 2. The performance gains of the oracle MMC on 5-shot binary classification tasks on various datasets. The derived MMC improves the few-shot performance of both metric and non-metric test-time methods: Nearest-Centroid Classifier (NCC) and Linear Classifier (LC).

Algorithm	Classifier	Transformation	mini	CUB	Texture	TS	PlantD	ISIC	ESAT	Sketch	QDraw	Fungi	Avg
PN	NCC	None	90.5	80.6	80.6	85.1	89.2	65.7	86.5	71.9	82.4	74.6	80.7
		Simple	91.3	82.4	83.1	85.8	93.0	68.6	89.2	75.2	85.1	77.2	83.1
		Oracle	**93.1**	**88.7**	**87.2**	**92.4**	**95.6**	**69.1**	**91.5**	**81.2**	**89.4**	**88.4**	**87.7**
S2M2	LC	None	94.0	87.1	85.7	88.7	95.0	68.7	93.5	78.7	85.5	82.8	86.0
		Simple	94.4	88.3	87.3	91.2	96.4	72.2	93.8	81.0	89.2	84.5	87.8
		Oracle	**96.3**	**94.0**	**90.7**	**96.1**	**98.3**	**72.6**	**95.2**	**87.0**	**93.0**	**93.3**	**91.7**

a clear impact of these properties on features is to make channel distribution smooth: suppress channels with high magnitude, and largely amplify channels with low magnitude. This phenomenon is clearly shown in Figure 2, where we plot Mean Magnitudes of feature Channels (MMC) on the test set of *mini*ImageNet and PlantDisease, with red ones being the original distribution, blue ones being the transformed distribution.

3.1. Deriving the Oracle MMC of Any Binary Task

We now wonder how much the MMC estimated by neural networks in a task deviates from the best MMC or *channel importance* of that task. We consider the binary classification problem. Specifically, let $\mathcal{D}_1$, $\mathcal{D}_2$ be the probability distributions of two classes over feature space $\mathcal{X} \subset \mathbb{R}^N$, and $\boldsymbol{x}_1 \sim \mathcal{D}_1$, $\boldsymbol{x}_2 \sim \mathcal{D}_2$ are samples of each class. Their means and covariance matrices are $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$, respectively. Then the original MMC of the binary task is defined as $\boldsymbol{\omega}^o = (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2$. We assume that the MMC after optimal adjustment is $\boldsymbol{\omega} \in \mathbb{R}^N$. We first standadize each channel l to have unit MMC, i.e., $\widetilde{x}_{i,l} = x_{i,l}/\omega^o_l, i = 1, 2, l = 1, .., N$. Then to adjust MMC to $\boldsymbol{\omega}$, the features should be transformed to $\boldsymbol{\omega} \odot \widetilde{\boldsymbol{x}}_1$ and $\boldsymbol{\omega} \odot \widetilde{\boldsymbol{x}}_2$. Here, we build a metric-based classifier. Specifically, a standardized feature $\widetilde{\boldsymbol{x}}$ is classified as the first class if $||\boldsymbol{\omega} \odot (\widetilde{\boldsymbol{x}} - \widetilde{\boldsymbol{\mu}}_1)||_2 < ||\boldsymbol{\omega} \odot (\widetilde{\boldsymbol{x}} - \widetilde{\boldsymbol{\mu}}_2)||_2$ and otherwise the second class. This classifier is actually the Nearest-Centroid Classifier (NCC) [14] with accurate centroids. Assume that two classes of images are sampled equal times. Then the average misclassification rate with this classifier is

$$\mathcal{R} = \frac{1}{2}[\mathbb{P}_{\boldsymbol{x}_1 \sim \mathcal{D}_1}(||\boldsymbol{\omega} \odot (\widetilde{\boldsymbol{x}}_1 - \widetilde{\boldsymbol{\mu}}_1)||_2 > ||\boldsymbol{\omega} \odot (\widetilde{\boldsymbol{x}}_1 - \widetilde{\boldsymbol{\mu}}_2)||_2) \\ + \mathbb{P}_{\boldsymbol{x}_2 \sim \mathcal{D}_2}(||\boldsymbol{\omega} \odot (\widetilde{\boldsymbol{x}}_2 - \widetilde{\boldsymbol{\mu}}_2)||_2 > ||\boldsymbol{\omega} \odot (\widetilde{\boldsymbol{x}}_2 - \widetilde{\boldsymbol{\mu}}_1)||_2)]. \tag{3}$$

Since multiplying a constant to $\boldsymbol{\omega}$ does not influence $\mathcal{R}$, we restrict $\boldsymbol{\omega}$ to have a fixed scale. We also assume that the channels are uncorrelated with each other. The following theorem gives an upper bound of the misclassification rate.

Proposition 3.1 *Given that* $\boldsymbol{\Sigma}_1 = \mathrm{diag}(\boldsymbol{\sigma}_1)$*,* $\boldsymbol{\Sigma}_2 = \mathrm{diag}(\boldsymbol{\sigma}_2)$*,* $\sum_{l=1}^{N} \frac{1}{\omega_l} = 1$*, we have*

$$\mathcal{R} < \sum_{l=1}^{N} \frac{2\omega_l(\sigma_{1,l} + \sigma_{2,l})^2}{(\mu_{1,l} - \mu_{2,l})^2} \tag{4}$$

To minimize this upper bound, the adjusted oracle MMC of each channel ω_l should satisfy:

$$\omega_l \propto \frac{|\mu_{1,l} - \mu_{2,l}|}{\sigma_{1,l} + \sigma_{2,l}} \tag{5}$$

We here use the word "oracle" because it is derived using true class statistics of the target dataset, which is not available in few-shot tasks. Table 2 shows the performance improvement over the simple feature transformation when adjusting the MMC to derived oracle one in each of the real few-shot binary classification tasks. The oracle MMC improves performance on all datasets, and always by a large margin.

3.2. Analysis of Channel Importance

Next, we take the derived MMC as an approximation of the ground-truth channel importance, and use it to observe how much of the channel emphasis of neural networks deviates from the ground-truth in each test-time dataset. We define MMC of a dataset D as the average l_1-normalized MMCs over all possible binary tasks in that dataset. Specifically, suppose in one dataset D there are C classes, and let $\boldsymbol{\omega}_{ij}$ denote the MMC in the binary task discriminating the i-th and j-th class. $\overline{\boldsymbol{\omega}_{ij}} = \boldsymbol{\omega}_{ij}/||\boldsymbol{\omega}_{ij}||_1$ normalizes the MMC, such that the l-th component of the vector $\overline{\boldsymbol{\omega}_{ij}}$ represents the percentage of channel emphasis on the l-th channel. Then the MMC of D is defined as $\boldsymbol{\omega}_D = \overline{\sum_{1 \leq i < j \leq C} \overline{\boldsymbol{\omega}_{ij}}}$, which gives average percentages of channel emphasis over all binary tasks. We visualize the oracle MMC, compared with MMC adjusted by the simple transformation and the original MMC of each dataset in Figure 3. To quantitatively measure the difference between different MMCs, we adopt the average normalized square difference $d(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{N}\sum_{l=1}^{N}(x_l - y_l)^2/x_l^2$; see Table 3.

Neural networks are overconfident in previously learned channel importance. Comparing the first and second rows in Table 3, we can see that the adjustment of MMC that the network made on new tasks is far from enough: the distance of original MMCs between train and test set (the first row) is much smaller than that between original and oracle MMCs on the test set. This suggests channels that are important to previously learned tasks are still considered by the neural network to be important for distinguishing entirely new tasks, but in fact, the discriminative channels are very likely to change on new tasks. This can be

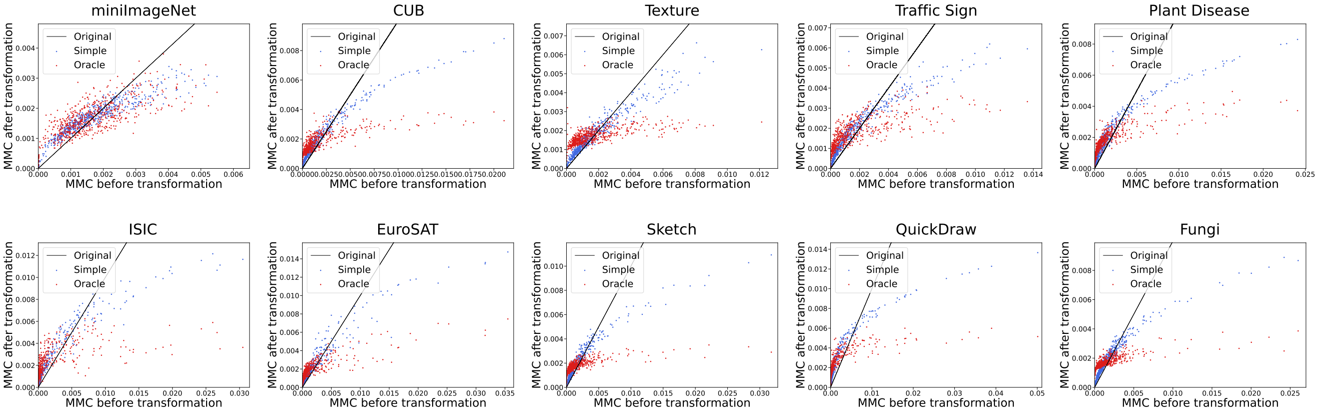

Figure 3. **Visualization of MMC of ten datasets ω_D before and after the use of simple and oracle transformation.** In each plot, a point represents a channel, and the x-axis and y-axis represent the MMC before and after transformation respectively, averaged over all possible binary tasks in the corresponding dataset. For comparison, we also plot the line $y = x$ representing the "None" scenario where none of the transformations are applied to features. The feature extractor is trained using PN on *mini*ImageNet.

Table 3. The average normalized square difference between different MMCs. The first row shows the distance between the original MMC of the training set (mini-train) and each test set; the second row shows the distance between the original and oracle MMCs on each dataset. The feature extractor is trained using PN on *mini*ImageNet.

		Test dataset										
Compared dataset	Trans.	mini-train	mini-test	CUB	Texture	TS	PlantD	ISIC	ESAT	Sketch	QDraw	Fungi
Train *v.s.* Test	None	-	0.18	1.56	0.88	1.13	1.54	2.28	1.30	1.01	1.58	0.79
Test	None *v.s.* Oracle	0.42	0.72	3.60	1.78	4.04	3.92	3.47	5.62	4.26	3.37	3.87

also observed from each plot in Figure 3, where the oracle MMC pushes up channels having small magnitudes and suppresses channels having large magnitudes. We call this the *channel bias problem* of neural networks.

The channel bias problem diminishes as task distribution shift lessens. The channel patterns in Figure 3 all look similar, except for *mini*ImageNet, whose overall pattern is close to the line $y = x$ representing the original MMCs. There does not exist *dominant* channels when testing on *mini*ImageNet (The maximum scale of channels is within 0.006), while on other datasets there are channels where the neural network assigns much higher but wrong MMCs which deviate far away from the $y = x$ line. In the second row in Table 3, we can also see that the distance between the original and oracle MMCs on *mini*ImageNet, especially on *mini*-train that the model trained on, is much smaller than that on other datasets. Since *mini*-test has a similar task distribution with *mini*-train, we can infer that the channel bias is less serious on datasets that have similar task distribution.

Channel bias distracts the neural network from new objects. In Figure 4, we compare some class activation maps before and after the oracle adjustment of MMC. We observe that adjusting channel importance helps the model adjust the attention to the objects responsible for classification using a classifier constructed by only a few support images. This matches observation in previous work [2, 17] that different channels of image representations are responsible for detecting different objects. The task distribution shift makes models confused about which object to focus on, and a proper adjustment of channel emphasis highlights the objects of interest.

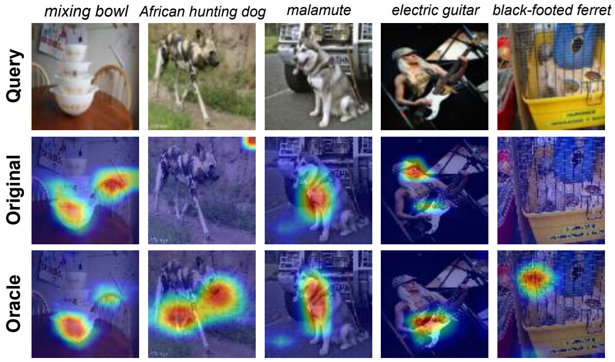

Figure 4. Some Grad-Cam [13] class activation maps of query samples using PN before and after the oracle adjustment of MMC on binary 5-shot tasks sampled from test set of *mini*ImageNet.

4. Conclusion

In this paper, we reveal the channel bias problem in few-shot learning, which is concerned with the difficulty of representation transfer of CNNs and needs further attention in the future. We show the problem can be alleviated by the presented simple channel-wise feature transformation.

References

[1] Mayank Agarwal, Mikhail Yurochkin, and Yuekai Sun. On sensitivity of meta-learning to support data. In *Advances in Neural Information Processing Systems*, 2021. 1

[2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017. 1, 4

[3] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: exploring simple meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9062–9071, 2021. 1, 2

[4] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations*, 2020. 1

[5] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *Advances in Neural Information Processing Systems*, 2020. 1

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

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2

[8] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019. 2

[9] Puneet Mangla, Mayank Singh, Abhishek Sinha, Nupur Kumari, Vineeth N. Balasubramanian, and Balaji Krishnamurthy. Charting the right manifold: Manifold mixup for few-shot learning. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2207–2216, 2020. 2

[10] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 2

[11] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. In *International Conference on Learning Representations*, 2020. 1

[12] Mamshad Nayeem Rizve, Salman H. Khan, Fahad Shahbaz Khan, and Mubarak Shah. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10836–10846, 2021. 1

[13] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 4

[14] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2, 3

[15] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In *European Conference on Computer Vision*, pages 266–282, 2020. 1

[16] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020. 2

[17] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations*, 2015. 1, 4