Texture Bias Of CNNs Limits Few-Shot Classification Performance

Sam Ringer[*] **Will Williams**[*] **Tom Ash** **Remi Francis** **David MacLeod**

Speechmatics
Cambridge, UK

{samr,willw}@speechmatics.com

Abstract

Accurate image classification given small amounts of labelled data (few-shot classification) remains an open problem in computer vision. In this work we examine how the known texture bias of Convolutional Neural Networks (CNNs) affects few-shot classification performance. Although texture bias can help in standard image classification, in this work we show it significantly harms few-shot classification performance. After correcting this bias we demonstrate state-of-the-art performance on the competitive *mini*ImageNet task using a method far simpler than the current best performing few-shot learning approaches.

1 Introduction

The ability of neural networks to perform image classification has increased dramatically in recent years [7, 4, 13]. However, much of this improvement has relied on using large amounts of labelled data, with successful classification of a given class often requiring thousands of labelled images for training. This is in contrast to the ability of humans to recognize new classes using only one or two labelled examples. The goal of *few-shot classification* is to bridge this gap such that machines can generalize their classification ability to unseen classes using very small amounts of labelled data.

[3] shows that Convolutional Neural Networks (CNNs) show a greater bias towards learning texture-based features than humans. However, they also demonstrate that this bias actually improves classification performance in the standard classification setting, where the classes at test time are drawn from the same distribution as those seen at train time.

At the time of writing there has been no investigation as to how the texture bias of CNNs affects classification performance when a distributional shift in classes is experienced between train and test time, such as that seen in the few-shot learning setting. This work performs this investigation and demonstrates that texture bias significantly decreases performance under distributional shift and correcting for this bias leads to large improvements in few-shot classification accuracy. We focus particularly on how the data *itself* can be manipulated or exploited to aid the learning process.

2 Related Work

2.1 Few-Shot Learning

A common methodology for evaluating few-shot image classification approaches is to pre-train a model on a corpus of labelled images and then test the model's classification ability on unseen

[*]Denotes equal contribution.

classes, given a small amount of labelled data from said unseen classes. The labelled data from the unseen classes is typically termed the *support* set and the images on which classification is tested is termed the *query* set. A wide range of approaches based on this methodology have been developed [2, 14, 11, 9].

One distinction between such approaches is that of *parametric* versus *non-parametric* methods. Parametric methods pre-train a model and, when presented with the support set, will *fine-tune* the parameters of the pre-trained model to improve performance on the query set. One such parametric approach is model-agnostic meta-learning (MAML) [2], which uses second-order gradients to learn an initialization that can be fine-tuned on a given support set. The resulting model can then perform classification on a corresponding query set. At the time of writing, the best performing parametric approach is classifier synthesis learning (CASTLE) [15], which synthesizes few-shot classifiers based on a shared neural dictionary across classes, and then combines these synthesized classifiers with standard 'many-shot' classifiers.

Non-parametric methods perform no fine-tuning when given the support set. Instead, they learn an embedding function and an associated metric space over which classification of new images can be performed. Such approaches include prototypical networks [12] and matching networks [14]. [12] learn an embedding function that maps images to points in a latent space. For a support set, each class 'prototype' is represented by the mean embedding of the examples in the support set. The query set is then classified according to the prototype that minimizes euclidean distance to the embedding of each query image. Non-parametric approaches have shown marginally inferior performance than parametric approaches. However, they are far simpler in their implementation at both at pre-training and test time.

2.2 Texture Bias Of CNNs

Outside of the few-shot learning field, there has been great progress in the interrogation of the features learned by CNNs when performing classification. Until recently, it was widely believed that CNNs were able to recognize objects through learning increasingly complicated spatial features [7]. However, [3] show that CNNs trained on the ImageNet dataset show extreme bias towards learning texture-based representations of images over shape-based representations. Furthermore, [8] show that shape is the single most important feature that humans use when classifying images.

[1] train CNNs with constrained receptive fields, effectively limiting the learned features to only low-frequency local features, such as texture. The resulting model achieves high test accuracies on ImageNet. This shows that texture-based features are adequate for good performance for standard image classification, where train and test classes are drawn from the same distribution.

Taken together, these results pose the question: why do CNNs learn to classify based almost entirely on texture where as humans rely mostly on shape? In this work, we consider a hypothesis: although high-frequency local features (such as texture) generalize well within a distribution, global low-frequency features (such as shape) generalize better under distributional shift. If this hypothesis were true, it could help explain why humans are so heavily dependent on shape-based features; it is because they generalize better to new classes than texture-based features.

3 Methods

3.1 Problem Formulation

In the typical few-shot classification formulation, a *task* consists of using a labelled *support* set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|S|}$ to classify an unseen *query* set $Q = \{\mathbf{x}_i\}_{i=1}^{|Q|}$. The support and query sets are sampled from the same set of classes. During the *pre-training* phase, tasks are sampled from a set of tasks $\mathcal{T}^{\text{pre-train}}$ and at test time the tasks are sampled from $\mathcal{T}^{\text{test}}$. There is no class overlap between $\mathcal{T}^{\text{pre-train}}$ and $\mathcal{T}^{\text{test}}$. A *k*-shot *n*-way task will contain images from *n* different classes and *k* images from each class. In this work we use the same training scheme, model and loss as [12].

3.2 Stylized Pre-training

[3] are able to remove the texture bias of CNNs by training on Stylized-ImageNet, which removes each image's texture by using AdaIN style transfer [5]. For this work, we create an analogous dataset: Stylized-*mini*ImageNet.

Our pre-training tasks are sampled from Stylized-*mini*ImageNet with probability p and from *mini*ImageNet with probability $1 - p$. By sampling from Stylized-*mini*ImageNet with a given probability, we can control the extent to which our model can learn texture-based features as opposed to shape-based features. At test time all tasks are sampled from the withheld classes of standard *mini*ImageNet.

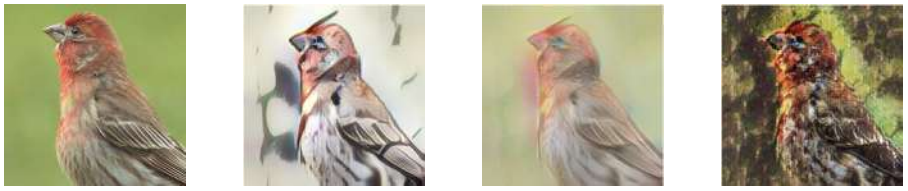

Figure 1: A sample from Stylized-*mini*ImageNet. The left-most image is the unstylized image with the remaining three being produced using different stylization kernels.

3.3 Support & Query Data Augmentation

At test-time, as the k-shot of a task increases, the problem tends from few-shot to standard many-shot classification and test accuracy increases dramatically. In light of this, we 'artificially' increase k-shot by performing data-augmentation on the support set. Each image in the support set is augmented $N_{Sa} = 32$ times. Our intention is to investigate the efficacy of emulating the high k setting using augmented examples from the support set. The prototype of class k, $\mathbf{c}_k$, is then given by:

$$\mathbf{c}_k = \frac{1}{|S_k| + N_{Sa} \times |S_k|} \sum_{(x_i, y_i) \in S_k} \left(f_\theta(\mathbf{x}_i) + \sum_{i=1}^{N_{Sa}} f_\theta(f_a(\mathbf{x}_i)) \right) \tag{1}$$

S_k is the support set for class k. f_a is a randomly sampled data augmentation function. f_θ is the learned embedding function, in our case a prototypical network [12].

For the query set, we also augment each image $N_{Qa} = 32$ times. Each of these augmented images is then passed through the embedding function and the estimated probability for a given original query image, $\mathbf{x}$, belonging to class k is given by equations 2 and 3.

$$p_\theta(y = k|\mathbf{x}) = \frac{\exp(-d(\mathbf{x}, \mathbf{c}_k))}{\sum_{k'} \exp(-d(\mathbf{x}, \mathbf{c}_{k'}))} \tag{2}$$

$$d(\mathbf{x}, \mathbf{c}_k) = \frac{1}{1 + N_{Qa}} \left(||f_\theta(\mathbf{x}) - \mathbf{c}_k||_2 + \sum_{i=1}^{N_{Qa}} ||f_\theta(f_a(\mathbf{x})) - \mathbf{c}_k||_2 \right) \tag{3}$$

4 Experiments

Table 1 shows the performance of our training scheme when testing on *mini*ImageNet for the 5-shot 5-way task. Our method is entirely non-parametric and far simpler to implement at both pre-train and test time than many of the other few-shot classification methods.

[3] show that training on a combination of unstylized and stylized data leads to a small drop in classification accuracy. This is because when the training and testing data are drawn from the same distribution (*i.e.* the same classes) the inherent texture bias of CNNs can actually aid performance.

However, in the few-shot learning setting, testing data is drawn from a different distribution to the training data. The ablation shown in Table 2 shows that pre-training on a combination of unstylized

Table 1: **Comparison to prior work on 5-shot 5-way *mini*ImageNet.** Conv-x denotes a 4-layer CNN with x filters in each layer.

model	**backbone**	**Test Accuracy**
Matching Networks [14]	Conv-64	55.3 ± 0.7
MAML [2]	Conv-32	63 ± 1
TADAM [10]	ResNet-12	76.7 ± 0.3
ProtoNet (Baseline) [12]	ResNet-12	76.8 ± 0.3
LEO [11]	WRN-28-10	77.6 ± 0.1
MetaOptNet [9]	ResNet-12	78.6 ± 0.5
CASTLE [15]	ResNet-12	79.52 ± 0.02
Ours	ResNet-12	80.4 ± 0.3

and stylized data actually *increases* performance at test time, even though the testing data is entirely unstylized. This suggests that the texture bias of CNNs does adversely impact performance under distributional shift. Figure 2 shows that as the proportion of Stylized-*mini*ImageNet pre-training

Table 2: **Ablation study.** The effects of pre-training on un-stylized and stylized data, as well as the effects of different forms of test-time augmentation.

Unstylized Data	**Stylized Data**	**Support TTA**	**Query TTA**	**Test Accuracy**
✓				76.8 ± 0.3
	✓			72.9 ± 0.3
✓	✓			78.8 ± 0.3
✓	✓	✓		79.2 ± 0.3
✓	✓	✓	✓	80.4 ± 0.3

data increases from 0 to 0.3, the accuracy increases as the resulting model is less biased towards texture-based features. However, as the proportion of Stylized-*mini*ImageNet increases from 0.3, the accuracy begins to decrease again as the final model is biased too heavily towards shape-based features.

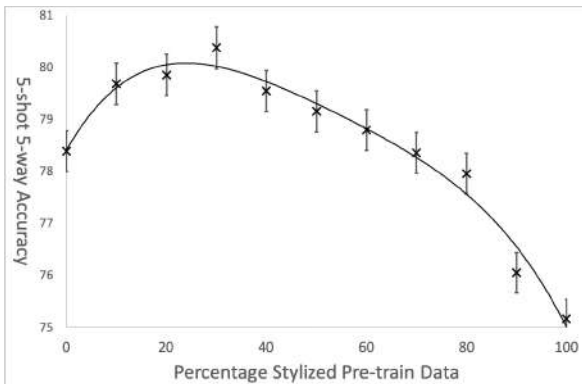

Figure 2: 5-shot 5-way *mini*ImageNet test accuracy versus pre-training data composition when using test-time augmentation.

5 Conclusion

It has previously been demonstrated that CNNs are biased towards learning texture-based features over the shape-based features used in human vision. Although this bias aids classification performance when training and testing classes are drawn from the same distribution (standard image classification), this work shows that the texture bias of CNNs significantly decreases classification performance in the few-shot learning setting, where distributional shift is experienced. Correcting for this bias achieves state-of-the-art performance on *mini*ImageNet, even using only a simple non-parametric method.

References

[1] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet, 2019.

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[3] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[5] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[8] Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive Development*, 3(3):299–321, 1988.

[9] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

[10] Boris N. Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. 2018.

[11] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. 2019.

[12] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. 2017.

[13] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[14] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, 2016.

[15] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Learning classifier synthesis for generalized few-shot learning, 2019.

Appendix

Experimental Setup

In this work we use a prototypical network [12] with a standard ResNet12 backbone [4]. For training we use the Adam [6] optimizer with an initial learning rate of 1×10^{-4} with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We do not initialize from any pre-trained weights and our model is trained for 70,000 steps, with the learning rate halving every 15,000 steps. We perform early stopping based off of validation accuracy. We use a temperature of 32 in the SoftMax function in equation 2. We test models over 20,000 sampled few-shot tasks, with the mean accuracy and 95% confidence intervals being reported above. We use no dropout, weight-decay or label smoothing.

Stylized-*mini*ImageNet

For the generation of Stylized-*mini*ImageNet, we began with *mini*ImageNet and used the same stylization procedure as [3]. [3] use a stylization coefficient of $\alpha = 1.0$ on ImageNet. When applied to the smaller images of *mini*ImageNet, this stylization coefficient led to images that were so distorted that even humans were unable to perform successful classification. For this reason, we generated Stylized-*mini*ImageNet with a less aggressive stylization coefficient of $\alpha = 0.4$.

To ensure diversity of styles and true independence of texture and underlying image, we generate 10 stylized images for each original *mini*ImageNet image. The stylization was performed only on the *train* split of *mini*ImageNet as testing and validation were both done on standard *mini*ImageNet.

Data Augmentation

At train and test time, we apply a standard set of data augmentations. The applied data augmentations are as follows: random horizontal flip, random brightness jitter, random contrast jitter, random saturation jitter and random crop between 70% and 100% of original image size. The final image is re-sized to be of size 84x84 pixels.