DATASET BIAS PREDICTION FOR FEW-SHOT IMAGE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

One of the obstacles which negatively affect the image classification performance is dataset bias. In particular, if each class has only a few training data samples, the data are highly likely to have dataset bias. Therefore, dataset bias can be a serious issue in few-shot learning, but has rarely been studied so far. To address this issue, we propose a bias prediction network to help improve the performance of few-shot image classification models. Once the features are extracted from an image data, the bias prediction network tries to recover the bias of the raw image such as color from the features. However, if the bias prediction network can recover it easily, we can assume that the extracted features also contain the color bias. Therefore, in our proposed framework, the full model tries to extract features that are difficult for the bias prediction network to recover from. We validate our method by adding the bias prediction network to several existing models and evaluating the performance improvement. Our experimental results show that the bias prediction network can suppress the negative effect of the dataset color bias, resulting in the substantial improvements in existing few-shot classification models. The proposed bias prediction network, which can be integrated with other models very easily, could potentially benefit many existing models for various tasks.

1 INTRODUCTION

Few-shot learning, in which a model is trained using only a few training samples in a training task, is a challenging topic of machine learning. A number of models has been proposed for few-shot learning (Koch et al., 2015), (Snell et al., 2017), (Sung et al., 2018), (Ren et al., 2018), (Rusu et al., 2019), (Lee et al., 2019), (Chu et al., 2019), (Li et al., 2019), (Zhang et al., 2020), (Medina et al., 2020). There are a lot of classes we want to classify, so methods for few-shot learning must be able to quickly adapt to new classes. Meta-learning (Lemke et al., 2015) can fulfill this requirement and thus, many recent methods have adopted meta-learning as an effective method for training and the classification of samples in few-shot learning.

Few-shot image classification is a subtopic of few-show learning. In each task, only a few classes are given for classification and only a few image data are given for each class. Actually to be a useful classification method, the method needs to classify many classes. Thus, a few classes are randomly selected from a given dataset in every task. Additionally, a few images are randomly selected from each class. The selected classes and images are changed at the start of a task. At the start of the training stage, the performance of the method will be poor. However as the training stage processes, the performance of the method will improve, despite using only a small amount of data.

There are many obstacles that reduce the performance of few-shot image classification. Here, we focus on one of these obstacles, dataset bias. Dataset bias means that some properties of the samples of a class in the training set are biased or not evenly distributed; thus, the model trained using this dataset may learn the properties in a biased manner. Figure 1 shows how the bias in a dataset could disturb classification. The photographs in the figure are from the *Adience* dataset (Eidinger et al., 2014). The dataset contains images of faces of people from different ages. Suppose that the classification problem is to classify the images into classes of young and old people. One of the physical characteristics of humans is the color of their skin. If a model is trained using images of only white-skinned young people (white circles) and black-skinned old people (black circles), both classifiers (solid line and dotted line) will be adequate to solve the problem. However, if the test

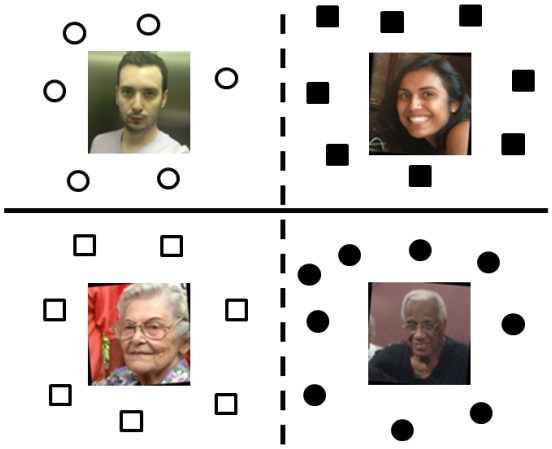


Figure 1: Problem of training a biased dataset. The figure represents a feature space. The white circles and the black circles are the training samples, and the white squares and the black squares are the test samples. We train a model to classify whether each face is young or old. The desirable classifier is the solid line, but if the model is trained to classify according to the dotted line, the classification result will be poor.

samples are composed of only black-skinned young people (black squares) and white-skinned old people (white squares), the solid line classifier is correct, but the dotted line is incorrect. In general, the greater the number of training samples, the lower the dataset bias in the samples. However, since only five or fewer training samples are typically used in a single task in few-shot image classification, dataset bias in the training samples is highly likely. If a model is trained using a dataset including bias, the model may learn the bias as important information. To prevent this problem, the model needs an additional mechanism that enables the model embed features that do not include the bias. To tackle this issue, we introduce a bias prediction network. We focus on the color bias as shown in Figure 1 in this study, but our approach is also applicable to other types of bias that can be represented as predictive targets. We train the bias prediction network to recover the bias of the raw image such as color from the embedded features. If the bias prediction network can almost recover the color bias, the embedded features are assumed to be highly dependent on the color components of the raw samples, which means that the features also have the color bias. If the model is trained to embed features that are difficult for the bias prediction network to recover from, the bias in the embedded features could be reduced. As a result, the general performance of the few-shot image classification will be improved.

The major contributions of this paper can be summarized as follows:

- We propose a bias prediction network to tackle the dataset bias issue in few-shot image classification. To the best of our knowledge, none of the previous studies have thoroughly investigated such issues.
- The bias prediction network is integrated into existing few-shot classification networks to predict the color bias and encourages the feature extraction module of the network not to learn the color bias. Therefore, the learning is done in an adversarial manner.
- The proposed network is compatible with and can be easily integrated with other models.
- Experiments show that the bias prediction network improves the performance of various existing few-shot classification models; thus, the proposed approach could potentially benefit many existing models for various tasks.

2 RELATED WORKS

2.1 FEW-SHOT LEARNING

Few-shot learning (FSL) is a challenging research topic where models learn a new concept or class from very few examples. An active subtopic of FSL is few-shot image classification (FSIC), in which images are classified using a method of FSL.

Generally, FSIC is performed by meta-learning. The FSIC model is trained using a chain of training tasks, and each training task contains a few data. The FSIC problem is called an N-way K-shot problem, where N is the number of classes (labels) and K is the number of data samples from each class. Every task is composed of a support set and a query set. The support set is used to learn to classify adequately. N classes are randomly selected and K data samples are randomly selected for each class. These $N \times K$ data form the support set. Additionally, to form the query set, some data are randomly selected from the N classes, but none of these data should be identical to any data from the support set. The query set is used to evaluate the performance on this task.

During training, the FSIC model extracts features from the given support set and generates classifiers. Then the model evaluates the performance on the data from the query set, and the model is updated depending on the performance. After a task is done, N new classes are randomly selected in the next task, and the evaluation and updating are repeated.

During testing, data from classes that are completely different from those selected for the training are used. Although the model has not learned these new classes before, the model can adequately classify the data.

The methods for FSIC models can be divided into two categories: distance-based methods and graph-based methods. Distance-based methods compare two feature vectors according to metrics. A Siamese network (Koch et al., 2015) extracts each feature vector from two images randomly selected from the support set and then compares the feature vectors using a trainable L1 distance. Matching Networks (Vinyals et al., 2016) also learn the distance function between the support vectors and the query vector. Prototypical networks (Snell et al., 2017) embed images to extract feature vectors and calculate the prototype vector for each class. Then when a feature vector is extracted from a query image, the image is predicted to be the class whose prototype vector is the nearest to the feature vector.

Graph-based methods include the graph neural network (GNN) model (Satorras & Estrach, 2018), in which each node represents the feature vector of each image and nodes are connected with edges. The similarities between neighboring nodes are calculated and used as the weight of the edge. The weighted average vector of the neighboring nodes is aggregated with the feature vector of the node. EGNN (Kim et al., 2019b) also utilizes the GNN architectures, and each edge between two nodes has the value of similarity between the two nodes. Then the values predict whether the two images belong to the same class.

More recently, methods focusing on classifiers have also been studied. MetaOptNet (Lee et al., 2019) utilizes linear classifiers trained using a linear support vector machine (SVM). DeepEMD (Zhang et al., 2020) has classifiers as subspaces and classify a feature vector by evaluating the distances from the feature vector to each subspace.

2.2 BIAS PREDICTION

Every dataset may have dataset bias such that the features of the data are not accurately represented. Dataset bias may lead to misclassification of the test samples. For example, if a dataset for facial analysis is largely composed of lighter-skinned subjects, error may occur when analyzing darker-skinned subjects (Buolamwini & Gebru, 2018).

Bias prediction is a method that predicts bias in a dataset and minimizes the bias to regularize data features. In a previous study for bias prediction (Kim et al., 2019a), the principle of adversarial learning (Goodfellow et al., 2014) was used. If the dataset bias exists, the labels can be predicted depending on the bias of the samples. This means that the mutual information between the bias of the training samples and labels will be large. Thus, the corresponding labels are closely related to

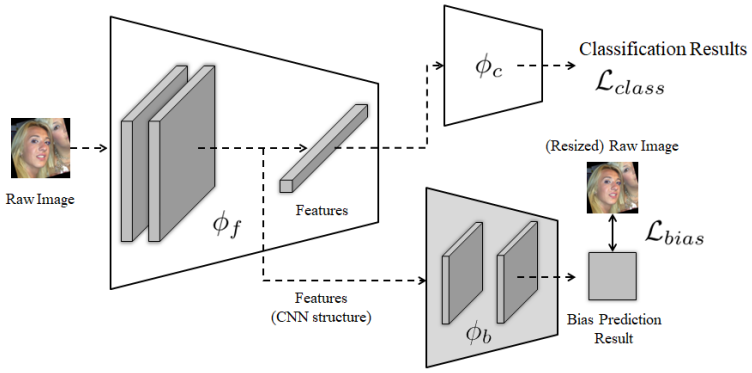


Figure 2: Overall architecture of neural networks with the bias prediction network. Generally an image classification model consists of a feature embedding network ϕ_f and a classification network ϕ_c. Here, the bias prediction network ϕ_b is following the feature embedding network. The bias prediction network gets the CNN-structured embedded features as input, and outputs bias prediction result. From the result, the bias prediction loss $\mathcal{L}_{bias}$ is obtained by calculating the cross-entropy between the resized raw image and the bias prediction result. The classification loss $\mathcal{L}_{class}$ is calculated by the original model, and used to calculate the total loss $\mathcal{L}_{total}$. The image samples are from *Adience* dataset.

the dataset bias. For instance, we assume that the dataset bias is the biased color distribution (e.g. human face skins in samples are all white or black). Then, we calculate the entropy of the embedded features from the feature embedding networks. If the bias prediction results represent a clear color distribution of the training samples, the entropy will be small. If a clear color distribution cannot be found from the bias prediction results, the entropy will be large. Therefore, the networks are trained to increase the entropy to reduce the effects of the dataset color bias. This network architecture is expected to achieve good performance despite the biased training dataset.

Since a few training data samples can hardly represent the color information of each class, we assume that the potential color bias must exist in the training data in FSIC problems. Thus, it is expected that our proposed bias prediction method can reduce the color bias and improve the performance of the original few-shot learning models.

3 Algorithm

3.1 The Bias Prediction Network

Suppose that we have been given the training set $\mathcal{X} = \{x_i, y_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ is the i-th image of the set and $y_i \in \mathcal{Y}$ is the corresponding label.

Generally, most of image classification models have two deep neural network modules: feature embedding and classifier. First, the feature embedding network extracts features from the raw image. Then the classifier obtains the features and outputs the classification result. The feature embedding network is denoted as ϕ_f, and the classifier is denoted as ϕ_c. Accordingly, the features, or the output of the feature embedding network can be written as $\phi_f(x)$, where x is a raw image data. The result of the classifier is denoted as $\phi_c(\phi_f(x))$.

We introduce the bias prediction network, denoted as ϕ_b. This network follows ϕ_f and takes the embedded features as input. If the features have the CNN-structure, the bias prediction network and the classifier both take $\phi_f(x)$ as input. Otherwise, ϕ_f should be modified to output CNN-structured features as well, since ϕ_b is implemented to take CNN-structured data as input. The bias prediction network ϕ_b is followed by a softmax function, and we need to define results both before and after the softmax. The bias prediction result before the softmax is written as $Z = \phi_b(\phi_f(x))$, and the result after the softmax is denoted as $\sigma(\phi_b(\phi_f(x))) = \sigma(Z)$, where σ is the softmax function.

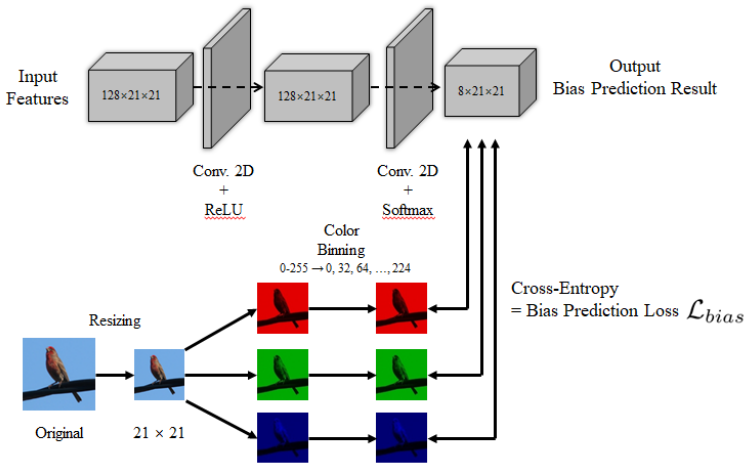


Figure 3: The detailed architecture of the bias prediction network ϕ_b predicting color bias. A raw image is resized and split into three channels, and then the value of each pixel is grouped into eight intervals to match the size to the output of the bias prediction network. Then the bias prediction loss is obtained by calculating the cross-entropy between the data of each channel and the output of the network.

We implement the bias prediction network with the algorithm introduced in (Kim et al., 2019a). Since we add a bias prediction network to the existing model at the output of the feature embedding network, the bias prediction network obtains embedded features as input and outputs the bias prediction result. The detailed structure of the bias prediction network is described in the next subsection.

We introduce two main loss functions in this algorithm; total loss and bias prediction loss. The total loss is denoted as $\mathcal{L}_{total}$, and the bias prediction loss is denoted as $\mathcal{L}_{bias}$. First, $\mathcal{L}_{total}$ is calculated and the original model (ϕ_f and ϕ_c) is updated, and then $\mathcal{L}_{bias}$ is calculated and ϕ_b is updated. This is one cycle of the training procedure for an image, and this cycle is repeated for other images.

3.2 The Total Loss

The total loss is given by

$$\mathcal{L}_{total} = \mathcal{L}_{class} - \lambda H\left(\sigma(Z)\right), \tag{1}$$

where $\mathcal{L}_{class}$ denotes the classification loss, $H(\cdot)$ denotes the entropy, and λ is the hyperparameter for the entropy regularization. Since the total loss is defined as above, the original feature embedding network ϕ_f is trained to make the entropy $H\left(\sigma(Z)\right)$ larger. The output $\sigma(Z)$ of the bias prediction network ϕ_b is for the color labels of the contracted image that the network recovers from the embedded features. Thus, if ϕ_b could expect each pixel of the contracted image easily, the entropy of the output would be low. However, if ϕ_b can easily expect the raw image from the embedded features, we can assume that the features highly depend on the color data of the raw image. Thus, ϕ_f tries to embed features that are barely dependent on the color distribution of the raw image.

3.3 The Bias Prediction Loss

Bias is most problematic when features do not represent the corresponding class labels in the dataset. For instance, hair color is not sufficient to represent cats and dogs, and a training dataset in which most cats have white hair and most dogs have black hair would lead to adequate classification of white cats and black dogs but not cats and dogs of other colors.

In this study, we predict the color bias of the raw images. Since the number of training samples is too small (less than or equal to five in most few-show learning), it is almost impossible to represent all the colors of each corresponding object. This means that a color bias must exist for each label. Thus, we structure the bias prediction network ϕ_b to recover the color components of the input image from the embedded features. To examine how many elements of the output $\sigma(Z)$ of the bias prediction

network ϕ_b recovered correctly, we compare the color labels of the resized raw image and the output of the network. Here, we use a cross-entropy function to define the bias prediction loss:

$$\mathcal{L}_{bias} = H\left(C, Z\right), \tag{2}$$

where $H(\cdot, \cdot)$ denotes a cross-entropy function and C is a matrix of true color labels, whose detailed description as follows.

Figure 3 shows the bias prediction network. Suppose the output of the feature embedding network has the size of $128 \times 21 \times 21$, the features are passed through the two CNNs and the bias prediction results are obtained with a size of $8 \times 21 \times 21$. To compare the bias prediction results with the true color labels, we match both sizes of the results and the labels. Thus, both the height and the width of the results are resized to 21 pixels. In addition, since the range of the values of the red, green, and blue channels is 0 to 255, if we apply color binning to each channel to split the range into eight equal intervals ($[0, 31], [32, 63], \cdots, [224, 255]$), then finally we get the color labels of the raw image that have the same size as the bias prediction results. The compressed data of the raw images are denoted by C.

The bias prediction loss is calculated by

$$\mathcal{L}_{bias} = \frac{\mathcal{L}_{bias}^{(red)} + \mathcal{L}_{bias}^{(green)} + \mathcal{L}_{bias}^{(blue)}}{3} \tag{3}$$

$$\mathcal{L}_{bias}^{(\cdot)} = \frac{1}{8 \cdot 21 \cdot 21} \sum_{i=1}^{8} \sum_{j=1}^{21} \sum_{k=1}^{21} c_{ijk} \log z_{ijk}, \tag{4}$$

where $\mathcal{L}_{bias}^{(red)}$, $\mathcal{L}_{bias}^{(green)}$, and $\mathcal{L}_{bias}^{(blue)}$ are the color bias prediction losses for the red, green, and blue channel, respectively, and c_{ijk} and $z_{ijk} (i = 1, \cdots, 8, j = 1, \cdots, 21, k = 1, \cdots, 21)$ are elements of C and Z, respectively.

3.4 Training Procedure

The original networks ϕ_f and ϕ_c are trained to minimize $\mathcal{L}_{total}$, whereas the additional bias prediction network ϕ_b is trained to minimize $\mathcal{L}_{bias}$. The classification loss $\mathcal{L}_{class}$ depends on the loss function of the original networks, and $H(\sigma(Z))$ is related to ϕ_b. On the other hand, $\mathcal{L}_{bias}$ is evaluated when ϕ_b predicts the color labels of the raw image from the embedded features.

In Eq. 1, since $\mathcal{L}_{total}$ becomes smaller if the entropy $H(\sigma(Z))$ is higher, ϕ_f tries to embed features that ϕ_b can hardly expect the color component of the raw image from. After ϕ_f is updated, ϕ_b tries to recover the color components of the raw image from the embedded features to minimize $\mathcal{L}_{bias}$. Then, ϕ_b is updated. This procedure is repeated for another training sample.

4 Experiments

In this section, we first explain the datasets used in our experiments and how we set the conditions of the network architectures. Next, we present the experimental results of applying the bias prediction network to different existing models and datasets.

4.1 Datasets and Implementation Details

We added the bias prediction network to the existing networks for few-shot classification, such as EGNN (Kim et al., 2019b), MetaOptNet (Lee et al., 2019), and DeepEMD (Zhang et al., 2020). We used the datasets *miniImageNet*, *CIFAR-FS*, *FC-100*, and *Adience* (Eidinger et al., 2014) for experiments.

To clearly illustrate the effect of the bias prediction network, we first created modified datasets from the *miniImageNet*. Figure 5a shows some examples of the modified datasets. From the original dataset, we created grayscale, red channel, green channel, and blue channel version datasets. Each of these datasets was used in the training stage, whereas the test set of the original dataset was used in the testing stage. Additionally, we tested the bias prediction network on the *Adience* face dataset,

Models	5-way 1-shot		5-way 5-shot	
	Original	BP Added	Original	BP Added
EGNN	46.68%	47.14%	61.50%	61.24%
MetaOptNet	54.68%	56.06%	71.07%	71.19%
DeepEMD	63.77%	64.57%	79.43%	79.99%

Table 1: Performances for the application of the bias prediction network on the *miniImageNet* datasets

Models	5-way 1-shot		5-way 5-shot	
	Original	BP Added	Original	BP Added
EGNN	51.69%	52.89%	64.86%	65.23%
MetaOptNet	59.58%	61.10%	73.61%	73.87%
DeepEMD	64.41%	65.88%	80.05%	80.42%

Table 2: Performances for the application of the bias prediction network on the *CIFAR-FS* datasets

where we trained the model with data from old black-skinned and young white-skinned people, and validated the model using samples from old white-skinned and young black-skinned people for age classification task.

The potential color bias in the dataset is expected to be larger when the size of the dataset is small. The *miniImageNet* dataset contains 600 images of each class. We also tested the network by using only 300 or 100 images per class.

4.2 Effect of the Bias Prediction Network

We tested the bias prediction network on several datasets and existing models for few-shot learning. We assumed the conditions of 5-way 1-shot and 5-way 5-shot learning as in most previous studies. We evaluated the accuracy depending on the application of the bias prediction network to each dataset and each existing model. Table 1, Table 2, and Table 3 show that in most cases, the bias prediction network improves the performance of the existing models.

4.3 Effect on Biased Datasets

We performed another test using the *Adience* dataset where the training samples and test samples were biased in the color of human skin. As shown in Figure 1, the training sample set consisted of white young people (white circles) and black old people (black circles), and the test sample set consisted of black young people (black squares) and white old people (white squares). We used EGNN as a base model, and tested the 2-way 5-shot learning problem of classifying young (approximately 25-32 years old) vs. old (approximately 60-100 years old) groups. The left part of Figure 4 shows the results. We verified that the original EGNN model was substantially improved byy the addition of the bias prediction network.

4.4 Effect on Color-Filtered Datasets

In this experiment, we tested the performance of the bias prediction network on the color-filtered *miniImageNet* datasets. To adjust the severity of the color bias, we split the raw images into red, green, and blue channels. Additionally, we desaturated the images to grayscale. Then, we evaluated the image classification results on each dataset. Some examples from the color-filtered *miniImageNet* dataset are displayed in Figure 5a. The base model was MetaOptNet, and we test under the 5-way 5-shot setting. Figure 5b shows the results. Compared to the models without the bias prediction network ($\lambda = 0$), the proposed model improved the performance on the red, green, and blue datasets. On the other hand, the bias prediction network did not improve the performance of the model on the grayscaled dataset. When images are grayscaled, red, green, and blue all have same

Models	5-way 1-shot		5-way 5-shot	
	Original	BP Added	Original	BP Added
EGNN	32.19%	33.55%	43.79%	44.07%
MetaOptNet	34.02%	34.72%	47.60%	47.93%
DeepEMD	41.23%	42.75%	53.43%	53.99%

Table 3: Performances for the application of the bias prediction network on the *FC-100* datasets

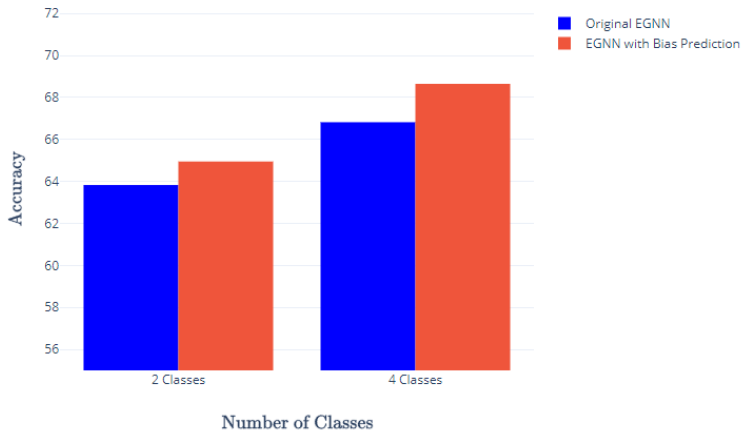


Figure 4: The results of the performance of the bias prediction network on the biased *Adience* dataset. EGNN was used as the original model. The horizontal axis represents the number of classes. The vertical axis represents the performance (accuracy) depending on the application of the bias prediction network. In both tests, the bias prediction network improved the performance of the original EGNN.

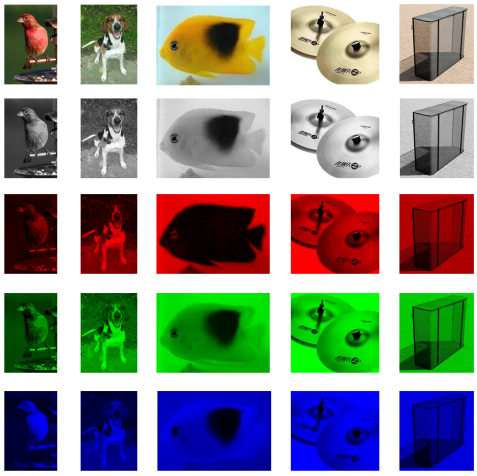
the value; thus, the color bias of the dataset is significantly reduced. Therefore, the bias prediction network barely improved the original model.

4.5 Comparison between Different Sizes of Datasets

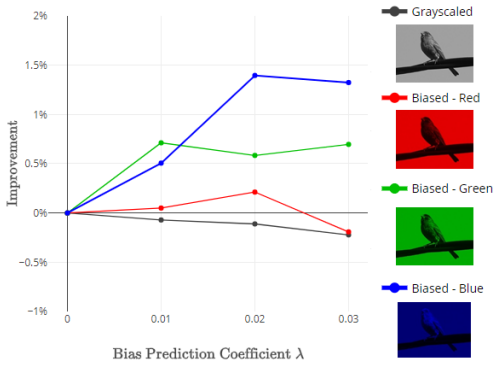
In this experiment, we verified the performance of the bias prediction network depending on the size of the *miniImageNet* datasets. The condition was 5-way 5-shot and the number of samples per class was set to 600, 300, and 100. Figure 6 shows the results. The results showed that the improvement of the performance was the greatest when each class contained 100 samples. This is since a fewer samples of each class can include larger potential color bias.

5 Conclusions

In this work, we tackled the problem of de-biasing extracted features from the existing models for few-shot learning and proposed a bias prediction network for color bias to improve the performance. Once the existing feature embedding network outputs features from the raw image, the bias prediction network tries to recover the color labels of the raw image from the embedded features. If the training set is color biased, then the existing model embeds features that are highly dependent on the color values of the training samples, and the bias prediction network can easily recover the raw image from the embedded features. Therefore, if the bias prediction network can almost recover the most part of the raw image, it is assumed that the embedded features are highly biased. Thus, we introduced a loss function to promote the ability of the existing model to embed features that are difficult for the bias prediction network to recover. Experimental results showed that our bias prediction network could improve the performance of various existing models. The proposed network is easy to integrate with various other models to provide potential benefits.



(a) The color-filtered versions of *miniImageNet* dataset. The first row shows the grayscaled version of the original examples. The second row, the third row, and the fourth row show the red, the green, and the blue channel images of the examples, respectively.



(b) Performance of the bias prediction network on the color-filtered *miniImageNet* dataset. MetaOptNet was used as the base model. The horizontal axis represents the bias prediction coefficient λ in Eq. 1. The vertical axis represents the improvement of the performance by the bias prediction network. The bias prediction network did not improve the performance on the grayscaled dataset, but did improve the performance on the red, green, and blue datasets.

Figure 5: Several examples of the color-filtered *miniImageNet* datasets and the experimental results for each dataset.

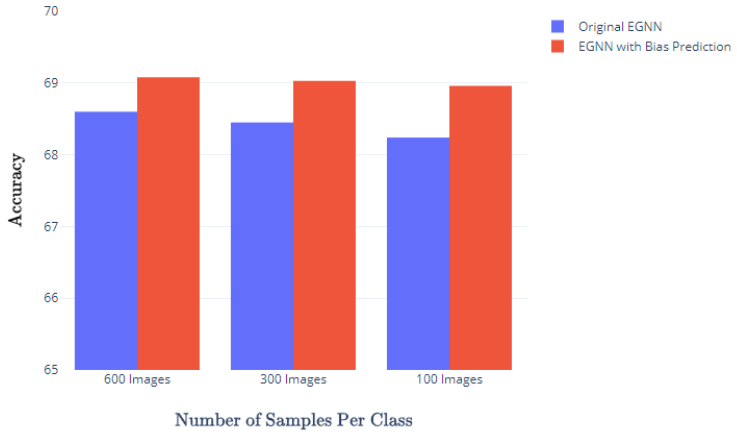


Figure 6: The results of the performance of the bias prediction network depending on the size of the *miniImageNet* dataset. EGNN was used as a base model. The horizontal axis represents the number of the samples of each class. The vertical axis represents the performance (accuracy) depending on the application of the bias prediction network. The lower the number of the samples used, the greater the effect of the bias prediction network.

REFERENCES

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pp. 77–91. PMLR, 2018.

Wen-Hsuan Chu, Yu-Jhe Li, Jing-Cheng Chang, and Yu-Chiang Frank Wang. Spot and learn: A maximum-entropy patch sampler for few-shot image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6251–6260, 2019.

Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9012–9020, 2019a.

Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019b.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Christiane Lemke, Marcin Budka, and Bogdan Gabrys. Metalearning: a survey of trends and technologies. *Artificial intelligence review*, 44(1):117–130, 2015.

Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/bf25356fd2a6e038f1a3a59c26687e80-Paper.pdf.

Carlos Medina, Arnout Devos, and Matthias Grossglauser. Self-supervised prototypical transfer learning for few-shot classification. *ArXiv*, abs/2006.11325, 2020.

Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJcSzz-CZ.

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJgklhAcK7.

Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJj6qGbRW.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1199–1208, 2018. doi: 10.1109/CVPR.2018.00131.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.

Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12203–12213, 2020.