Biases are Features: Unbiased Learning on Unknown Bias

Myeongho Jeon
Seoul National University
Seoul, Republic of Korea
andyjeon@snu.ac.kr

Daegyung Kim
Seoul National University
Seoul, Republic of Korea
sunnmoon137@snu.ac.kr

Woochul Lee
Seoul National University
Seoul, Republic of Korea
woochukee@snu.ac.kr

Joonseock Lee
Seoul National University
Seoul, Republic of Korea
joonseock@snu.ac.kr

Abstract

Convolutional Neural Networks achieve the state-of-the-art in image classification task. However, their dependence on training data distribution makes encoded representation partial on biased condition. Existing unbiased training methods need to define bias on multi-variant sensory datasets. To address this issue, we propose a new framework that does not need to know which the bias is in given training dataset. We find biased classifier is especially vulnerable when they are overly dependent on biased features. Under this observation, the proposed framework refers hierarchically different features from stacked convolution layers. They are transmitted to calculate discriminative confidence via orthogonally regularized operations. This framework enforces the model to be dependent on multi-level features and so be robust on biased distribution. Extensive evaluation on newly constructed feature dataset and intentionally biased dataset demonstrate our proposed framework is effective to unbiased learning. To contribute to future research, code of this work is available on https://github.com/aandyjeon/biases-are-features

1. Introduction

As the field of applying machine learning algorithms increases, the importance of robustness in machine learning is also increasing. As a branch of research for robustness, unbiased training methods have recently been dedicated [5, 11, 2, 1] . The model could be biased because the learning process of the machine learning algorithm is highly dependent on the training data distribution [22, 23]. Although a well-distributed dataset can reduce the bias of the model, it is difficult to define and construct. The biases are learned as meaningful features on biased data distribution, which makes the model unrobust in real-world. To train the model unbiasedly, a clear answer to *Which is the bias?* is required.

In an image classification task, Geirhos et al. [5] and Banhg et al. [2] consider texture as bias. Kim et al. [11] define color or age as bias. Adeli et al. [1] consider shade on face and gender as bias to be unlearned. They show robust results for several biased datasets by reducing the use of bias in the training process. However, their proposed framework defines bias and minimizes the dependency on it, which is limited in following twofold: (i) Different distributions for different datasets make it difficult to define bias. And previously undefined bias could induce a biased model. (ii) Although a bias can be employed as an important feature in prediction, a model gives it up for unbiased training. To address these issues, we do not define bias and assume all the biases are features. We use both biased and unbiased features for training, but regularize the model to avoid being overly dependent on few features.

According to [5], the ImageNet trained CNN-based classification model is highly dependent on texture. Inspired by [5], we experiment feature dependency of state-of-the-art models on more various features from several datasets. It is confirmed in the performance gap of the pre-trained model between the original test set and feature images in test set. In our experiment, class-specific features (*e.g.* ear, eye, nose) is considered as an important feature while color, silhouette and edge are not. A model trained too dependently on a few features are vulnerable to real-world data in which these features are absent (see Figure 1).

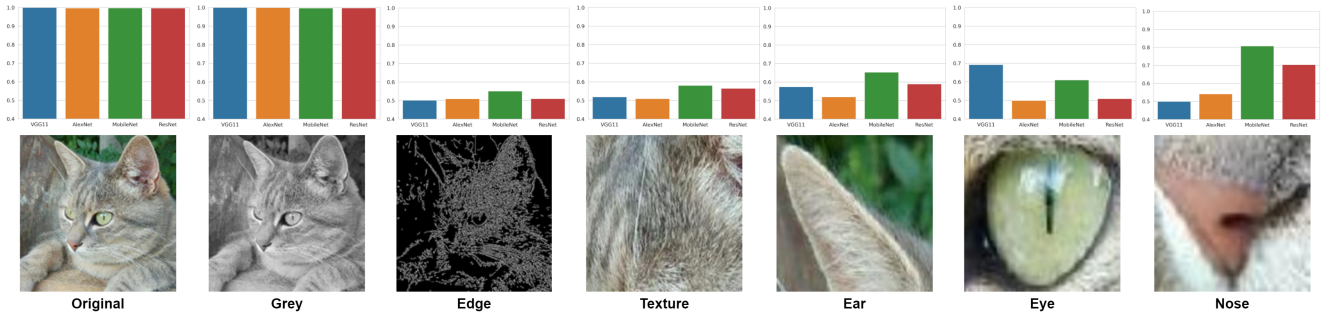

Figure 1. Performance gap between biased test sets. Each of the X-axis of the graph denotes VGGNet(blue), AlexNet(orange), MobileNet(green), and ResNet(red) respectively. Y-axis denotes the accuracy on Oxford-IIIT Pet Dataset which is composed of dog and cat images for classification. Note that in binary classification, the accuracy of 50% means the model almost does not work.

In the process of training sequentially stacked layers in a classification model, each layer shows hierarchical nature of the features [25]. For instance, shallow layers capture corners and edge/color conjunctions. And deep layers capture significant variation and class-specific features. In this regard, low-level features and high-level features can be considered as subsets of the entire feature set that make up the input images. We use feature maps from different layers to regularize that the trained model is dependent on a small number of features. Our main contributions can be summarized as follows:

1. Our experiment on various feature images from several dataset indicate that previous image classification framework is highly dependent on few features, which makes the model vulnerable to the biased dataset.

2. We propose a new framework for unbiased training that employ intermediate feature maps sequentially generated from CNN-based model. With this framework, bias is not needed to be defined because unbiased other features can be exploited for prediction.

3. We present a feature dependency estimating protocol, which is the performance gap between highly biased test sets.

2. Related Work

2.1. Robustness

A test image to which an imperceptible non-random perturbation is applied often change the network's prediction [21, 18, 6, 17, 26]. Although deformations of input during training increase the robustness [12], they are limited to cover different distribution from original dataset such as adversarial examples [21]. To address adversarial examples, a line of works is dedicated.

Deep neural networks are often incapable of correctly assessing the uncertainty in the training data and so make overly confident decisions about the correct class, prediction or action [3]. In real-world decision making systems, however, classification networks must not only be accurate, but also should indicate when they are likely to be incorrect [7, 10]. For this issue, works to evaluate predictive uncertainty are dedicated [13, 19].

Modern neural networks are known to generalize well when the training and testing data are sampled from the same distribution. However, when deploying neural networks in real-world applications, there is often very little control over the testing data distribution [15]. To reduce performance gap between different distribution, a line of works are dedicated [8, 14, 9].

2.2. Bias Removal on Image Classification

The bias removal protocol has recently become a major branch of robustness in deep learning. Kim et al. [11] employ an additional network to predict the bias distribution and train the network adversarially against the feature embedding network. They formulate regularization loss based on mutual information. Geirhos et al. [5] show that CNNs trained on ImageNet are strongly biased towards recognising textures rather than shape variation, which is in stark contrast to human behavioural evidence.They construct Stylized-ImageNet which make the model be able to learn shape-based representations. Bahng et al. [2] encourage de-biased representation to be different from a set of representations that are biased by Hilbert-Schmidt independence criterion. They assume that the model can be intentionally biased towards texture by reducing the receptive fields. Although HEX [24] is proposed to address domain adaption problem, it can be applied to bias removal task. The authors of [24] quantify texture bias by utilising the neural gray-level co-occurrence matrix. The biased features are encouraged to be removed through the projection in the learned representations. Adeli et al. [1] define surrogate loss for predicting the bias while quantifying the statistical dependence with respect to target bias based on squared Pearson correlation.

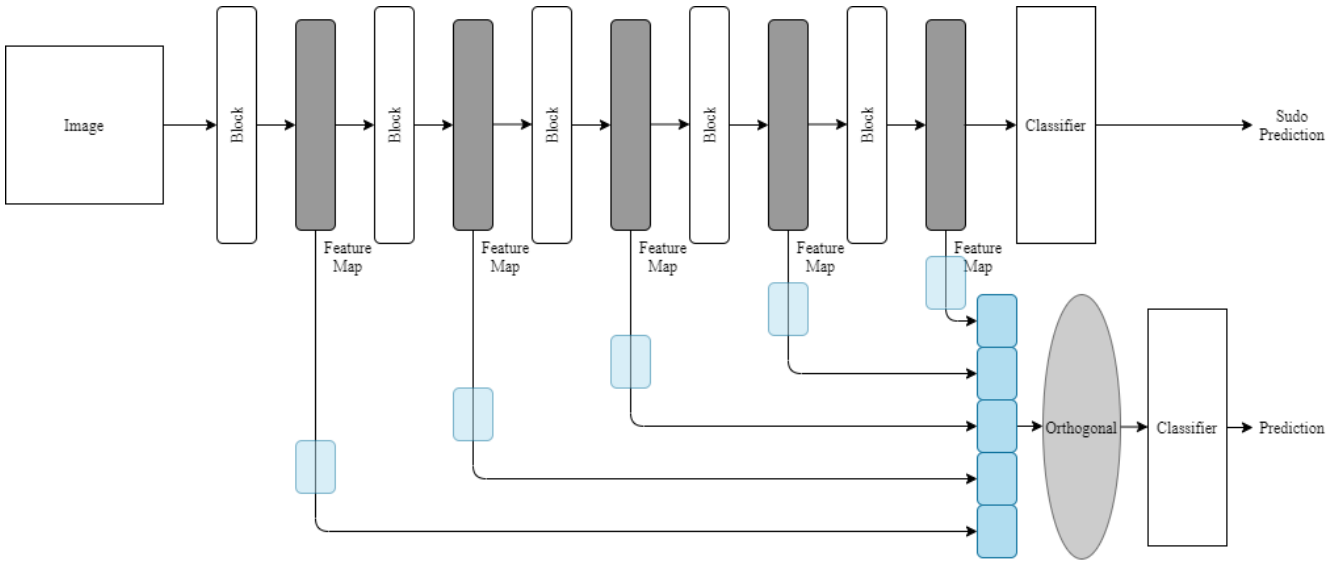

Figure 2. Overall architecture of our proposed model

3. Problem Statement

We define an image as a set of features $X = \{x_1, x_2, x_3, \dots\}$, where x_i denotes the features (e.g. color, texture, shape, ...). Biased model problem can be expressed as follow:

$$\mathcal{I}(b(X_{train}); Y) \gg \mathcal{I}(b(X_{test}); Y) \approx 0, \quad (1)$$

$$\mathcal{I}(b(X); D(X)) \gg 0, \quad (2)$$

$$b(X) \subset \{x_1, x_2, x_3, \dots\}, \quad (3)$$

where X_{train} and X_{test} denote the random variable sampled during the training and test procedure, respectively, $\mathcal{I}(a; b)$ denotes the mutual information between a and b, $D(X)$ denotes the discriminative network and $b(X)$ denotes the bias of X. Biased training data results in the biased networks, so that the network relies heavily on the bias of the data. Note that Biases defined in previously dedicated works [5, 11, 2, 1] are age, color, texture, which could be also considered as features.

The training procedure is to optimize the following problem:

$$\min_{\theta_D} E[L_c(Y, D(X))] + \lambda \mathcal{I}(b(X); D(X)) \approx 0, \quad (4)$$

where $L_c(a, b)$ denotes cross-entropy loss, and λ is a hyper-parameter to balance the terms.

In this paper, we aim to regularize our model to employ multiple features in input image and so robust on biased dataset. To estimate the robustness of our model on feature-absented test set, the performance gap between X and the subset of X (e.g. $X - x_i$ or x_i) is evaluated. To estimate the robustness of our model on biased test set, experiments on biased dataset such as colored MNIST are exploited.

4. Proposed Method

Our goal is to design a framework to regularize the model not to be highly dependent on certain minority features. According to the analysis in [25], we assume the features made hierarchically from low to high-level by sequentially staked layers could be helpful for the model robustness. Our proposed model consists of a baseline network, which is a sequentially stacked convolutional neural network such as VGGNet, and a main network that makes final predictions utilizing multiple features. (see Figure 2) The baseline network extracts features, and the main network is trained to exploit these features effectively. In order to perform their respective roles independently, the backpropagation steps of the two networks are performed independently with the weight of the other network being frozen. All the feature maps from different blocks of baseline network are employed for prediction (See Figure 2). We define repeating layers as a block (A conv-conv-pool-conv-conv-pool network consists of two conv-conv-pool blocks). The set of feature maps F can be expressed as:

$$F = \{f_1, f_2, f_3, \dots, f_n\}, \quad (5)$$

where f_i denotes feature maps extracted from the baseline network sequentially. Note that all the f_i are different sizes.The feature maps should be same size to be concatenated by 1×1 convolution operation and global averaging pooling. With the results squeezed spatially and channel-

Figure 3. Animal Faces-HQ dataset feature images. From left are original, gray, edge, texture, eye, and nose images.

wisely, the concatenated outputs are as below,

$$\begin{aligned} G_A &= [g_1, g_2, g_3, \dots, g_n] \\ G_{0,1} &= [0, g_2, g_3, \dots, g_n] \\ &\dots \\ G_{0,i} &= [g_1, g_2, g_3, \dots, 0, \dots, g_n] \\ &\dots \\ G_{0,n} &= [g_1, g_2, g_3, \dots, 0], \end{aligned} \tag{6}$$

where $g_i \in \mathcal{R}^{N \times N \times C}$ denotes squeezed feature from each f_i, G_A and $G_{0,i}$ denote the result from all of the extracted features and the set of features excluding one feature respectively, 0 denotes a padding matrix with all zeros, and [;] denotes concatenation. Please note that $G_{0,i}$ is used for projection in Section 4.2

4.1. Orthogonal Network

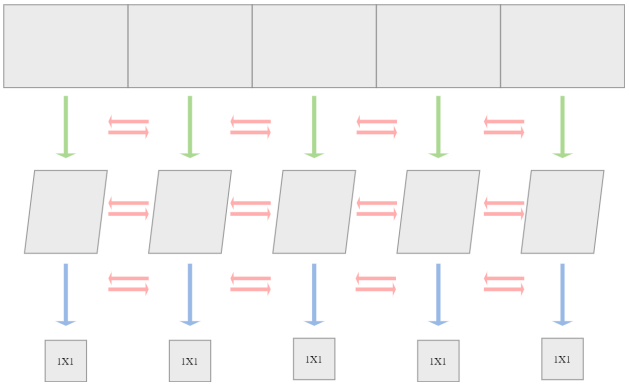

Figure 4. Orthogonal network

Even if hierarchically different features from different blocks are transmitted independently, they can still be combined to make predictions dependent on a small number of features. In particular, operations that share weights between channels, such as convolutional layers and fully connected layers, are vulnerable to this issue. Inspired by [16], we exploit orthogonal convolution and orthogonal squeeze block to maintain low to high-level features for final prediction (See Figure 4).

Main network receives concatenated features $G_A \in \mathcal{R}^{n \times N \times N \times C}$ as input and outputs confidences for prediction. Orthogonal convolution is n-grouped convolution of which weights are regularized by orthogonal loss. Orthogonal convolution can be formulated as:

$$\begin{aligned} C_{conv} &= \frac{W_{conv} \cdot W_{conv}^T}{\|W_{conv}\| \times \|W_{conv}^T\|} \\ L_{orth} &= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} C_{conv}(i,j), \end{aligned} \tag{7}$$

where W_{conv} denotes n-grouped convolution weight, $C_{conv} \in \mathcal{R}^{n \times n}$ denotes cosine similarity matrix between weights.

Following squeeze operation is applied n-grouped features independently with orthogonal loss between weights one another. Squeeze operation can be expressed as:

$$F_{squeeze} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} W_{squeeze}(i,j) \times G(i,j) \tag{8}$$

where $W_{squeeze} \in \mathcal{R}^{W \times H}$, of which W and H are feature maps's spatial size, denotes squeeze weight and $F_{squeeze} \in \mathcal{R}^{1 \times 1 \times channels}$ denotes the output of squeeze operation. With n numbers of $W_{squeeze}$, objective function for orthogonality can be expressed as:

$$
\begin{aligned}
W_{concat} &= [W_{squeeze1}, W_{squeeze2}, \ldots, W_{squeeze3}] \\
C_{squeeze} &= \frac{W_{concat} \cdot W_{concat}^{T}}{\|W_{concat}\| \times \|W_{concat}^{T}\|} \\
L_{orth} &= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} C_{squeeze}(i,j),
\end{aligned} \tag{9}
$$

where [;] denotes concatenation, $C_{squeeze} \in \mathcal{R}^{n \times n}$ denotes cosine similarity matrix between weights.

4.2. Projection

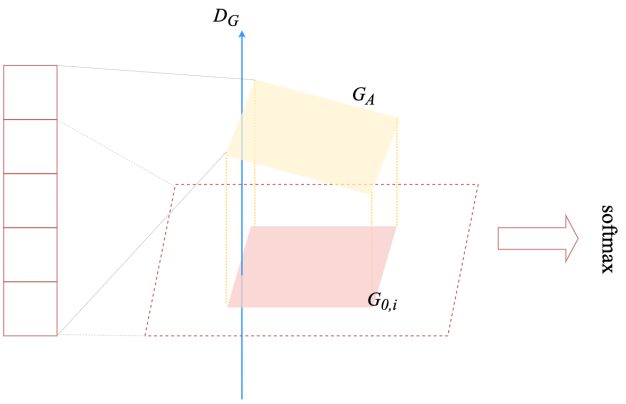

Figure 5. Projection

Although feature maps obtained from sequentially stacked layers are different features of the image, there may be some overlaps between these feature maps. This make the model biased on much-overlapped features. We assume features can be expressed as sum of multiple directions on representation space. Projecting all the concatenated features to some sub space enforce model to exclude features on sub space (See Figure 5). With this operation, low to high-level features are discriminated once again. Inspired by [24], we apply projection method to select only completely independent features. The projection operation can be formulated as follow:

$$
G_{p,i} = (I - G_{0,i}({G_{0,i}}^{T} G_{0,i})^{-1} {G_{0,i}}^{T}) G_A \tag{10}
$$

$$
F_A = [G_{p,1}, G_{p,2}, G_{p,3}, \ldots, G_{p,n}] \tag{11}
$$

$$
\hat{y} = softmax(FC(F_A)), \tag{12}
$$

where G_{pi} denotes the projected matrix of G_A by G_{0i}, and $softmax$ and FC denote softmax activation layer and fully connected layer, respectively.

5. Experiments

In this section, we present dataset we utilized, implementation and results.

5.1. Dataset

In this paper, we utilize Animal Face-HQ (AFHQ) dataset [4] and colored MNIST [11]. Among dog, cat, and wild in AFHQ dataset, we employ dog and cat images because various wild animals have inconsistent features. To evaluate feature dependency of proposed model, We construct feature subset of X (*e.g.* $X_i, X - X_i$ in Section 3). For gray and edge feature images, we apply gray scaling and canny edge algorithm respectively. Texture and class-specific features (ear, eye, nose) are cropped manually to extract exact location. As AFHQ dataset consists of images $X \in \mathcal{R}^{512 \times 512}$, cropped feature images have enough semantic information for discrimination (see Figure 3).

Colored MNIST is a intentionally biased dataset planted color bias into the MNIST dataset. ten distinct colors are assigned to each digit category as their mean color. each color of digit are sampled from normal distribution of corresponding mean color and given variance. For variance σ^2, from 0.02 to 0.05 with a 0.005 interval are applied.

5.2. Implementation

In the following experiments, we evaluate performance gap between gray, edge, texture, and class-specific feature images on AFHQ dataset. And colored MNIST dataset is exploited to evaluate the effectiveness of our proposed method on intentionally biased dataset.

We use VGGNet11 [20] as baseline network architecture for AFHQ dataset and plain network with four convolution layers for colored MNIST experiments. VGGNet11 is pre-trained with ImageNet data and fully connected layer is added for binary classification. For plain network, every convolution layer is followed by batch normalization and ReLU activation layers.

5.3. Results

AFHQ dataset we compare our model to baseline network VGGNet11 performance gap between feature images of AFHQ dataset to evaluate whether our proposed framework reduces feature dependency variance. all the performances below 0.5 are rounded up to 0.5. In our experiment on orthogonal network, the accuracy of highly dependent feature ear is reduced, and that of edge is increased (See Table 2). Additional ablation study demonstrates orthogonality between weights helps generalization ability of model (See Table 3).

Colored MNIST We compare our proposed framework with other methods that can be used for this task. Bias removal model can be categorized to supervised model, which needs bias as explicit label, and unsupervised model. Note that although explicit labels are not needed for HEX [24] and rebias [2], they are different framework from our model in that they are designed to unlearn pre-defined bias such as

Method	Baseline (baseline) Unsupervised	rebias Unsupervised	Projection(our) Unsupervised	Orthogonal (our) Unsupervised	Kim et al. [11] Supervised
0.02 (σ^2)	0.4000	0.4966	0.4856	0.5169	0.6700
0.025	0.5486	0.5723	0.6363	0.5803	0.7100
0.03	0.6422	0.7005	0.6737	0.6792	0.7800
0.035	0.6896	0.7838	0.7771	0.7725	0.8150
0.04	0.7628	0.8297	0.8417	0.8064	0.8600
0.045	0.7964	0.8733	0.8697	0.8709	0.8900
0.05	0.8448	0.8837	0.9009	0.9242	0.9200

Table 1. Experiment on colored MNIST. All the results are the average of three experiments.

color, texture. HEX [24] is excluded in our experiment because they use gray-scaled images as input, which remove color bias. Our projection and orthogonal method outperform baseline network on all the dataset and state-of-the-art on some deviations (See Table 1). Additional ablation study demonstrates orthogonality between weights is effective to unbiased learning (See Table 4).

test set	baseline network	orthogonal network
original	1.0000	0.9990
gray	1.0000	0.9985
edge	0.5010	0.5613
texture	0.5190	0.5048
ear	0.6924	0.6241
eye	0.5000	0.5008
nose	0.5740	0.5183

Table 2. Feature dependency on AFHQ dataset

Orthogonal		✓
gray	0.999	0.999
edge	0.500	0.561
texture	0.507	0.505
ear	0.589	0.624
eye	0.593	0.501
nose	0.512	0.518

Table 3. Ablation study on feature dependency test. The number of parameter and operation of comparison model is same as our proposed model. Grouped convolution of orthogonal network is replaced to convolution operation and orthogonal loss is not employed.

6. Conclusion

In this work, we consider that discriminative model is vulnerable to biased datasets when trained dependent on certain minority features. To address this issue, we propose a framework that exploits multi-level features hierarchically created by sequential convolutional layers. hierarchical features are maintained until the final prediction through orthogonal convolution and squeeze block. This framework

Orthogonal		✓
0.02(σ^2)	0.457	0.517
0.025	0.547	0.580
0.03	0.644	0.679
0.035	0.729	0.773
0.04	0.782	0.806
0.045	0.827	0.871
0.05	0.866	0.924

Table 4. Ablation study on colored MNIST dataset. The number of parameter and operation of comparison model is same as our proposed model. Grouped convolution of orthogonal network is replaced to convolution operation and orthogonal loss is not employed.

does not need to define bias, which is more appropriate than existing unbiased learning methods in real-world. Extensive experiment on newly constructed feature evaluation set demonstrate our proposed framework has even distribution of performance. Discrimination based on multiple features is confirmed to be robust on biased dataset. We hope that this framework will add value to the future research.

References

[1] Ehsan Adeli, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Li Fei-Fei, Juan Carlos Niebles, and Kilian M Pohl. Representation learning with statistical independence to mitigate bias. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2513–2523, 2021. 1, 2, 3

[2] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pages 528–539. PMLR, 2020. 1, 2, 3, 5

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015. 2

[4] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 5

[5] 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. *arXiv preprint arXiv:1811.12231*, 2018. 1, 2, 3

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. 2

[8] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. 2

[9] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018. 2

[10] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017. 2

[11] 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*, pages 9012–9020, 2019. 1, 2, 3, 5, 6

[12] 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. 2

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016. 2

[14] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017. 2

[15] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017. 2

[16] Bingchen Liu, Yizhe Zhu, Zuohui Fu, Gerard de Melo, and Ahmed Elgammal. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4836–4843, 2020. 4

[17] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015. 2

[18] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 2

[19] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019. 2

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[21] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[22] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *Domain adaptation in computer vision applications*, pages 37–55. Springer, 2017. 1

[23] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. 1

[24] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. *arXiv preprint arXiv:1903.06256*, 2019. 2, 5, 6

[25] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2, 3

[26] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4480–4488, 2016. 2