Simple Interpretation-Based Data Augmentation Method for Mitigating Shortcut Learning

Zizhe CHEN
Chinese University of Hong Kong
1155173938@link.cuhk.edu.hk

Abstract

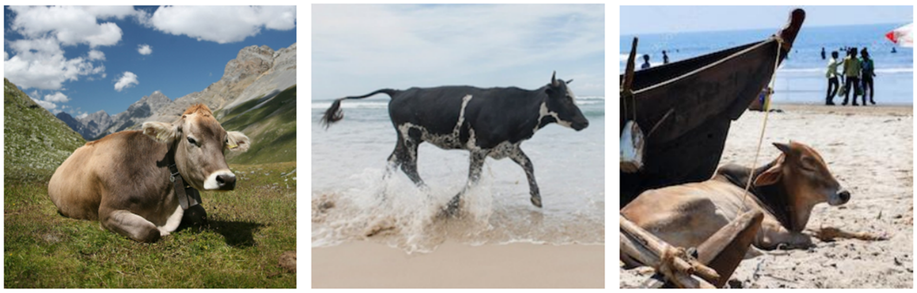
Despite brilliant success of deep learning, there are also many weakness, some of which might be fatal to the performance. In this report, we study one of the underlying problems of deep learning: shortcut learning. Most of the previous methods for detecting and eliminating shortcut learning require prior knowledge about sensitive features, as well as additional training and data collection process, which can be expensive and tedious to perform. Therefore, we propose the Interpretation-Based Data Augmentation Method (IBDAM) to mitigate the shortcut learning of the neural network. IBDAM is simple and intuitive, without requirement of any prior knowledge or data collection. By applying IBDAM to the ideal test examples constructed by us that contain severe shortcut learning, we observe an improvement 10% to 20% in prediction accuracy. Also, we also extend the IBDAM to chest X-Ray (CXR) dataset and implement the shorT method to evaluate its performance. Finally, we analyze the limitation and future improvement of IBDAM.

1 Introduction

Deep learning has triggered the current rise of artificial intelligence and numerous success stories have rapidly spread, but behind the success of deep learning, there is many weakness and limitation. This paper focuses one of the weakness of deep learning: shortcut learning, which refers to phenomenon that neural network model acquires high accuracy in the independent and identically distributed (i.i.d) dataset through learning undesired sensitive features[3]. Figure 1 is a concrete example of shortcut learning in object recognition task. In the first image, there is cow on the pasture, and neural network recognizes the cow with 0.99 probability. However, in the second and third image which contain cow on the beach, neural network fails to recognize the cow with high probability. The reason is that, instead of being based on the feature and shape of the cow, the prediction of cow is based on the pasture. This is what the general schematic of shortcut learning entails in Figure 2. Cow is a condition. Whether it exists or not will affect content of image. The pasture and beach are attributes, which affect both image and the condition. What we want is to predict condition based on its feature in image, but the neural network predicts condition based on the attribute.

Shortcut learning is observed in many areas of deep learning, such as computer vision, natural language processing, and healthcare[3]. Although the model using shortcut learning achieves high accuracy in the i.i.d. dataset, e.g. cow on the pasture, when tested in the out-of-distribution (o.o.d.) dataset, e.g. cow on the beach, the performance always degrades dramatically[3]. Therefore, shortcut learning has seriously endangered the general performance of many learning tasks, such as classification, contrastive learning, and few-shot learning[1 13 8]. Shortcut learning is the result of dataset bias, and the shortcut learning itself can be interpreted as some kinds of bias (for concise of report, "bias" and "shortcut learning" are use interchangably in this report). Previously, researchers have propose many excellent methods for detecting and eliminating shortcut learning. There are some supervised methods to identify and eliminate shortcut learning with prior knowledge of the sensitive features of

Preprint. Under review.



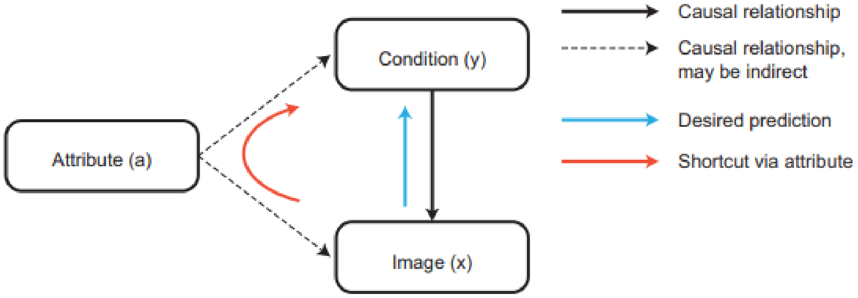
(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97

(C) No Person: 0.97, **Mammal: 0.96**, Water: 0.94, Beach: 0.94, Two: 0.94

3

Figure 1: A concrete example of shortcut learning in object recognization task.



1

Figure 2: A schematic of shortcut learning.

shortcut learning, and additional training and data collection is required[3 1]. There are also other high level automated methods to reduce shortcut learning, including training an extra adversary network or performing modification on the high-level feature[13 10]. These methods are described in detail in Section 2.

According to our understanding, there is no research to implement and evaluate some simpler and more intuitive ideas for mitigating shortcut learning automatically. Therefore, in this project, we try to propose a simpler, automatical, and general method to mitigate shortcut learning. Since data augmentation is an important way to improve generality of deep learning[15] and some interpretation methods can help us gain knowledge about how neural network makes prediction[14 12], it is possible to combine these two simple ideas together to prevent neural network from learning shortcut features. Based on that, we propose an interpretation-based data augmentation method (IBDAM) to automatically mediate shortcut learning phenomenon. This method does not require prior knowledge of sensitive features or collection o.o.d data or too much extra computational cost. Its idea is simple and intuitive, so it can be easily modified and adapted to other tasks.

Contribution of this report:

1. Propose, describe, and implement the IBDAM method.
2. Design theoretical experiment with MNIST dataset and practical experiment with CIFAR10 dataset to verify the effectiveness of IBDAM.
3. Extend the IBDAM method to chest X-Ray dataset (CXR) and implement shorT in Pytorch to check the performance of IBDAM. Analysis the limitation and propose future improvement direction.

2 Related Work

Shortcut learning: A body of prior work studied the features that trigger shortcut learning as well as detection and solution of shortcut learning. By adding different types of noise and features to the images in the training data,[9] shows that unlike human vision, which biases the overall shape of the object, the neural network does not bias the overall shape, and it has a strong learning ability to learn and leverage any correlated features or noise within a category. Following the path of[9], I designed more experiments about the behavior of the neural network. To solve shortcut learning,[3] proposed a method of constructing o.o.d dataset, which requires prior knowledge of sensitive features and collecting data.[1] proposed another method of multitask learning to measure and control the strength of shortcut learning, but this method also requires prior knowledge of sensitive features and tedious evaluation through model training. Since features in image are implicit, it's too expensive for human to analyze every attribute. There are also automated methods that are high level, for example,[10] trained an extra 'lens' network to make small changes to the image to remove shortcut features in self-supervised learning, and[13] proposed implicit feature modifications to modify the high-level feature to suppress shortcut solutions in contrastive learning. Contemplating about these methods, I find that the simple and intuitive idea of IBDAM was not implemented and evaluated before. Therefore, I propose IBDAM and apply it to various dataset to evalute its performance.

Interpretation: Given model's predictions, interpretation methods can generate heatmaps, which highlight the region that is most related to the prediction. Currently there are many different interpretation methods. Randomized Input Sampling for Explanation (RISE) is a general interpretation method that can be incorporated into any black-box model. It chooses a certain number of pixels to mask with Monte Caro sampling and calculates the expected change in the prediction value to derive the importance of each pixel[12]. Although RISE has good generality, it is too costly because multiple model inferences are required to obtain one interpretation, so that it is not applied in IBDAMC. urrently, one of the most popular methods for interpretation is Grad-CAM, which interprets model prediction by backpropagate the prediction to the last layer of neural network and calculate importance of each region by weighted sum of activation of channels[14]. Since only part of backpropagation is required, it greatly reduce the time cost of IBDAM. There are many other interpretation methods based on Grad-CAM such as Eigen-CAM and LayerCAM, which have different performance on different images and tasks[11][5]. They will be applied and tested in IBDAM.

Data augmentation: IBDAM use data augmentation techniques to twist the shortcut features so that neural networks cannot learn them easily. Since the learning of the position shortcut can be effectively solved by randomly cropping the image[3], IBDAM focuses on data augmentation of the pixels value. To twist the shortcut features, the most intuitive choice is the Random Erasing and Cutout[18][2], and they indeed perform well in the experiment. The idea of IBDAM is similar to the idea of interpretation-based cutout raised in[2], but the authors only tested the idea in the i.i.d. dataset and did not dive into this idea.[17] also propose a similar method of Interpretability-Mask, which is different from IBDAM and focuses on masking out the background unrelated to the object. What's more, to control the degree of distortion of feature, IBDAM is also experimented with the Gaussian Patch proposed by[7], but the experiment result shows that, in clean data, Cutout is better than Gaussian Patch.

3 Data

There are three datasets, which are chosen according to different image types and difficulty, for the test, experiment, and evaluation of IBDAM. The first dataset is MNIST, which consists of 60000 handwritten numbers. It will be used to create 3 different sub-datasets shown in Figure 3. The first sub-dataset is the biased train dataset. It contains only two classes with "0" image as one class and "1" image as another class. For the "0" class, the number 0 is always in the bottom-right corner. For the "1" class, the number 1 is always in the top-left corner. The second sub-dataset is unbiased train dataset , in which there are also "0" class and "1" class, and the position of number 0 and 1 is randomly distributed in the image. The third sub-dataset is test dataset, in which the position of number 0 and 1 is opposite to the biased train dataset: The number 0 is always in the top-left corner and the number 1 is always in the bottom-right corner.

The second dataset is CIFAR-10. Since the train data and test data of CIFAR-10 are sampled from the i.i.d. dataset, it is used to preliminarily implement and experiment IBDAM in the i.i.d. dataset.

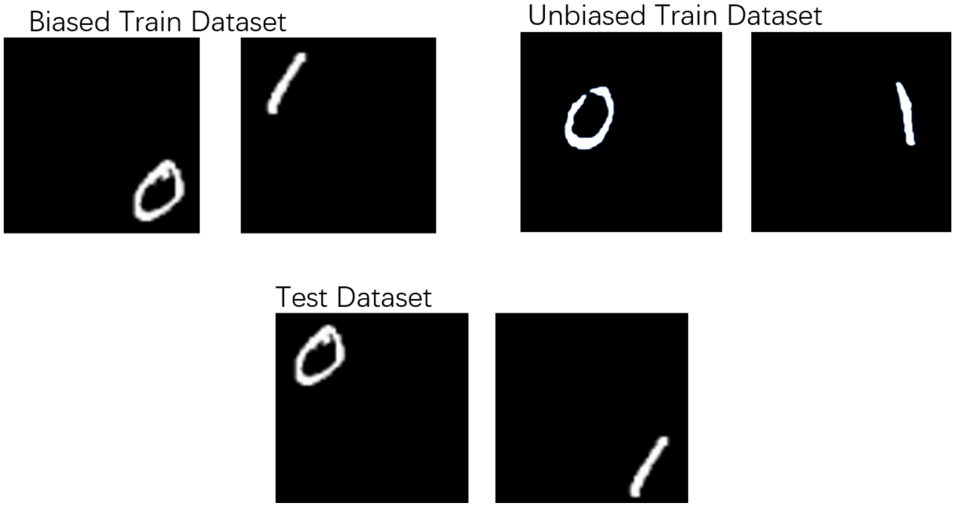


Figure 3: The biased train dataset.

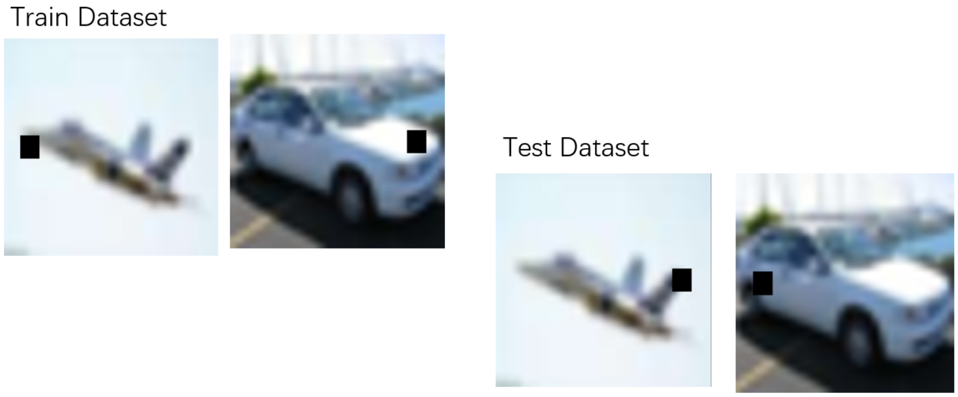


Figure 4: The ideal shortcut sub-dataset. The black patch on the images of train dataset is shortcut that manually added. In the test dataset, the location of black patch is opposite to the train dataset.

In addition, to evaluate the performance of the IBDAM method, we also sample some images from the CIFAR-10 dataset and construct ideal shortcut sub-datasets, for example, as Figure 4 shows, containing only two classes of plane and car, one sub-dataset adds a black patch to the images and different patch position in different location of different classes is the shortcut for neural network. In the test dataset of this sub-dataset, the position of the black patch in opposite to the train dataset, so we can measure the degree of shortcut learning by evaluate the test loss and accuracy.

The third dataset that we use is the NIH Chest X-RAY(CXR) Dataset from the NIH Clinical Center (source: https://nihcc.app.box.com/v/ChestXray-NIHCC), which is comprised of 112120 X-ray images with disease labels from 30805 unique patients. One example of NIH CXR is Figure 2. Current research shows that the original NIH CXR dataset has bias on age, which can trigger shortcut learning, and more tricky shortcut learning on patient's age can be triggered by sample distribution with larger age bias from the dataset[1]. We will extend the IBDAM method to this dataset and use the shorT method[1] to evaluate its performance. Based on the evaluation result from NIH CXR dataset, we further discuss the limitation as well as future improvement of IBDAM method.

4 Approach

Idea of IBDAM is intuitive. Shown in Figure 6, this method can be divided into three steps. First step is training with basic data augmentation, since IBDAM requires valid interpretation of model

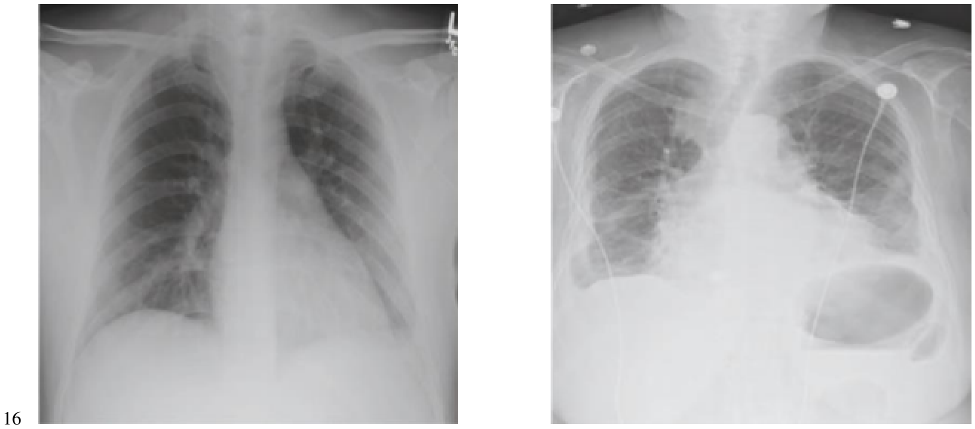


Figure 5: Two image samples from the NIH CXR dataset.

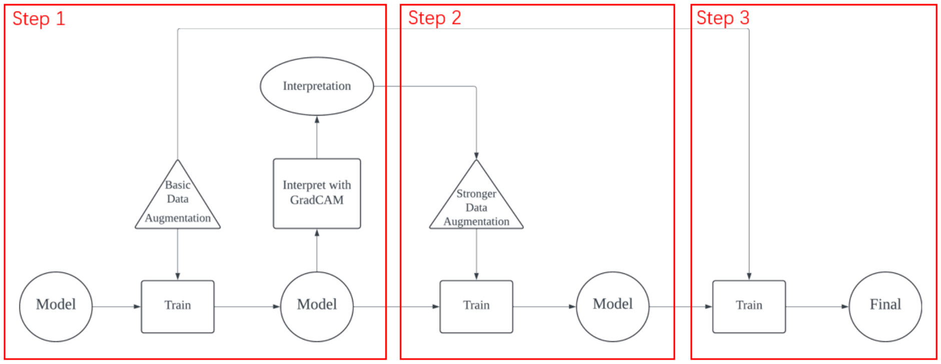


Figure 6: A schematic of the IBDAM method. The ’Model’ always refers to the trained model from last step, and the arrow can be interpreted as ’apply to’ relation.

prediction. The model is trained on the dataset with only the basic data augmentation of Horizontal Flip, Random Crop, and Normalization for 100 epochs or until converge.This basic data augmentation is always applied in the later steps if there is no specification. After training, the model can make some predictions for Grad-CAM to interpret. Then, for each image in the training data (no basic augmentation), use Grad-CAM to generate a corresponding heatmap, which is pure heatmap without overlapping with image (see Figure 7), and store the heatmap. The extra time cost for is equivalent to the training time of one epoch. The storage space is 20 to 100 times smaller than the original dataset, because majority of the heatmap is single white, grey, or black color. What’s more, with a proper encoding of the output of Grad-CAM, the storage cost will be even smaller. When it comes to the second step, the model continues to train with the weight obtained in the first step training for 75 epochs or until converge. During training, both the training data and its corresponding heatmaps are loaded. According to the different highlight of different region, extra data augmentation policy with different intensity is applied. Until now, I have experimented with several simple data augmentation policies, among which the Cutout performs best and I will only present the experiment with Cutout in Section 5 for simplicity. In theory, the extra time cost of the third step depends on the size of the image and the specific data augmentation policy, for example, for size equal to 224*224 and data augmentation policy is Erasing, the extra time cost is only 5 to 10 seconds. It’s worth noting that to make the model fully learn other features, the heatmaps are not changed in the whole step 3. After step 2 the extra data augmentation is removed, and the model continues to train for 25 epochs or until converge in step 3. The intuition and reason for step 2 and step 3 of IBDAM are explained in Section 5.

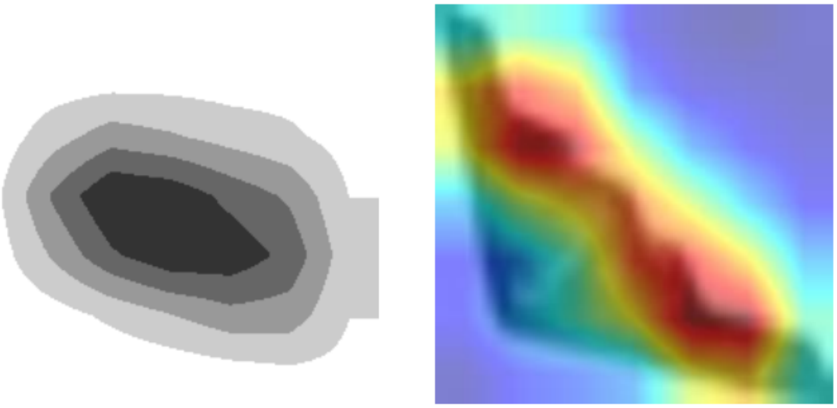


Figure 7: The pure heatmap compares with heatmap that overlaps with the image.

5 Experiments

5.1 Experiments on MNIST

In this section, I will show some important experiment result. All the model used in experiment is 18-layer Residual Net[4] implemented in Pytorch. We use SGD optimizer with default settings. The batch size is 64 and the training stop when the train loss and accuracy is converge. To ensure consistence between different dataset, images from all the dataset are resampled to size 224*224.

As we promise in Section 4, now, we show some experiments of the behavior of neural networks on the MNIST dataset to give intuition and reason of step 2 and step 3 of IBDAM. First experiment is Trigger experiment, the model is trained on the biased train dataset and test on the test dataset demonstrated in Figure 3 to see whether the shortcut learning of position can be triggered. We repeat this experiment for ten times. Each time, when the train loss converges to 0, with the test accuracy always approaches 0, the result is that the shortcut learning is always triggered.

Second is the Recover experiment. The model is trained with biased train dataset and then trained on the unbiased train dataset with random position of 1 and 0 in Figure 3. The goal is to see whether the model can recover from shortcut learning and converge to the unbiased dataset, which is decided by test accuracy on test dataset of Figure 3. With final test accuracy always approaches 100, the result is that model can recover from the shortcut learning.

Third is the Corrupt experiment. The model is trained with unbiased train dataset with random position of 1 and 0 and then trained with the biased train dataset. The goal is to see whether the model without shortcut learning will be corrupted by the dataset with obvious shortcut learning pattern, which is decided by the test accuracy on test dataset. After being trained on biased train dataset for more than 50 epochs, the test accuracy is still over 90. Therefore, non-biased will not be easily corrupted by the shortcut learning pattern.

Although the above three experiments are simple, they reveal an important insight of the behavior of neural network while facing shortcut learning: biased model can recover from shortcut learning, but non-biased model will not be corrupted by shortcut learning pattern. We assume that neural network can roughly follow this rule in all other tasks. This is the intuition behind the step 2 and step 3 of IBDAM. Since biased model can recover from shortcut learning, in theory, in the step 3, where the extra data augmentation is removed, model can recover from any shortcut learning introduced by extra data augmentation of step 2. Since non-biased model cannot be corrupted by shortcut learning pattern, the shortcut learning corrected by step 2 cannot be reintroduced in step 3.

5.2 Experiments on Ideal Shortcut Dataset

Then, with the insight revealed, We will show the real pipeline of applying IBDAM through the following experiment on ideal shortcut sub-dataset sampled from CIFAR10 shown in Figure 4. In step 1, we train the model with this ideal shortcut sub-dataset. Figure 8 shows the plot of test accuracy

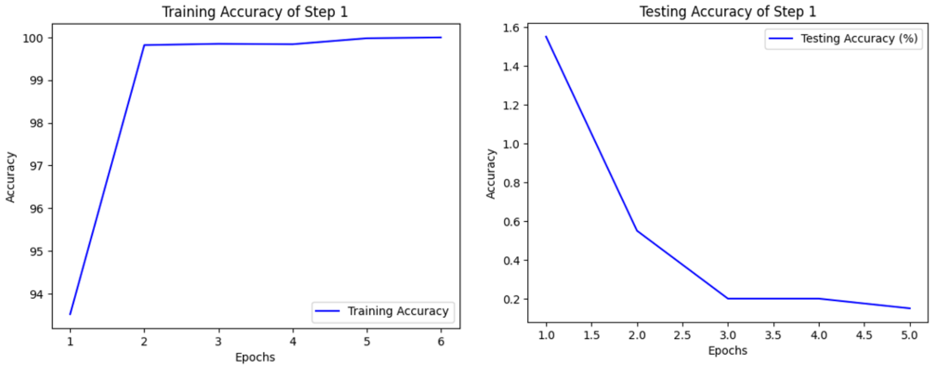


Figure 8: The plot of test accuracy and train accuracy in step 1.

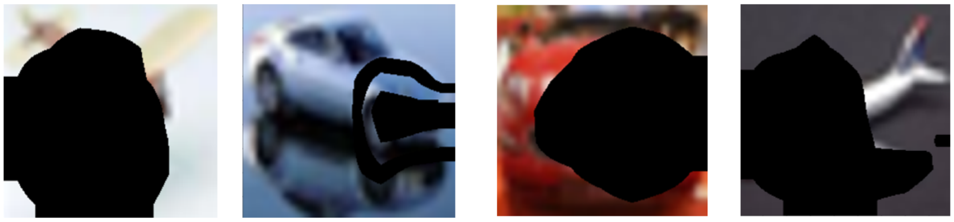


Figure 9: The images after being applied with stronger data augmentation in step 2.

and train accuracy during model training in step 1. When the train accuracy converge to 100%, the test accuracy approach to 0%, which means severe shortcut learning has happened during training.

In step 2, we apply stronger data augmentation based on the interpretation. For simplification, we apply Cutout as the stronger data augmentation. The images after being applied with stronger data augmentation is shown in Figure 9. Obviously, the feature that neural network depends on to perform shortcut learning, the black patch, is blocked and polluted by the stronger data augmentation. The training result of step 2 is shown in Figure 10. With the train accuracy converges to 95%, the test accuracy is around 50%. Although the test accuracy is even no better than the accuracy of random guess, at least compared with the 0% test accuracy in step 1, there is great improvement, which means that neural network can learn some desired feature after using stronger data augmentation method to block the shortcut learning label.

In step 3, we remove the stronger data augmentation and train the model with basic data augmentation until converge. Figure 11 show the plot of train accuracy and test accuracy. With train accuracy converge to 100%, the test accuracy is around 25%. Although the accuracy is even worse than the accuracy of random guess, at least compared with the 0% accuracy in step 1 and compare with the result of applying other data augmentation techinques in Table 1, the accuracy is improved and the shortcut learning is mitigated. It is also important that the 25% test accuracy means neural network does not experience the catastrophic forgetting[9] and it still remembers some useful features learned

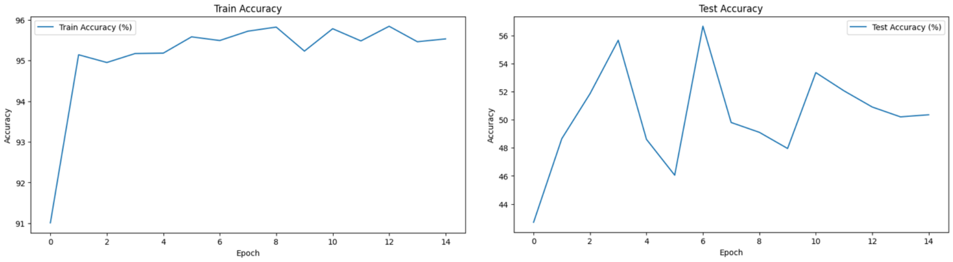


Figure 10: The plot of test accuracy and train accuracy in step 2.

Table 1: Effectiveness of Different Data Augmentation Method on Ideal Shortcut Dataset

Methods	Cutout	AutoAugmentation	Crop	IBDAM-gaussian	IBDAM-Cutout
Train Accuracy(%)	100	100	100	100	100
Test Accuracy(%)	0.75	0.5	0.85	**17.75**	**25**

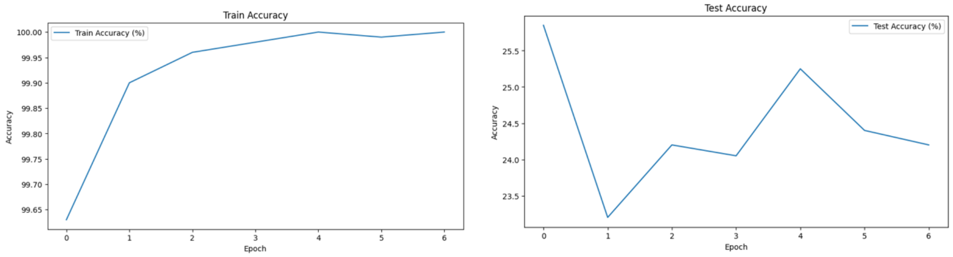


Figure 11: The plot of test accuracy and train accuracy in step 3.

in the step 2. I believe that with well design data augmentation methods, the performance of this IBDAM has great potential for improvement.

5.3 Experiments on Unbiased Dataset

Besides mitigating shortcut learning, IBDAM also has a good property that, if it is applied to dataset and model without bias and shortcut, it will not affect the accuracy of the model. We experiment this with a model trained on the original CIFAR10 dataset and the result is shown in the Figure 12. The first step is from epoch 0 to 100 and it is a normal training process. In step 2, the stronger data augmentation is applied and the performance of model inevitably decreases because large amount of features is blocked. When it comes to step 3, as soon as the stronger data augmentation is removed, the model retains the performance in step 1 and sometimes obtain better performance as Table 2 show.

5.4 Experiments on NIH CXR Dataset and Limitations

Finally, we further extend IBDAM to the NIH CXR dataset, which contains the shortcut learning on the attribute of age to check its performance in different dataset. However, different from the Ideal

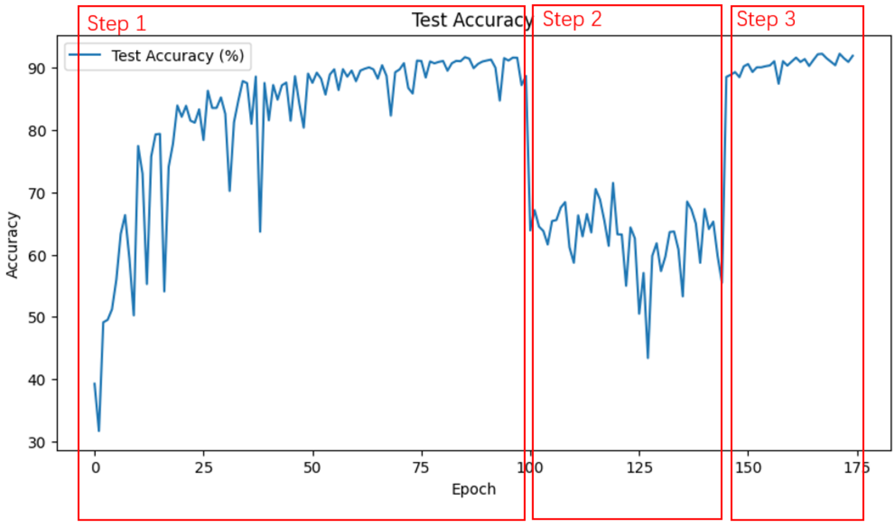


Figure 12: Train model with IBDAM on unbiased dataset CIFAR10.

Table 2: Effectiveness of Different Data Augmentation Method on Unbiased Dataset

Methods	Cutout	AutoAugmentation	Crop	IBDAM-gaussian	IBDAM-Cutout
Test Accuracy(%)	93.38	92.71	91.64	**92.22**	**93.63**

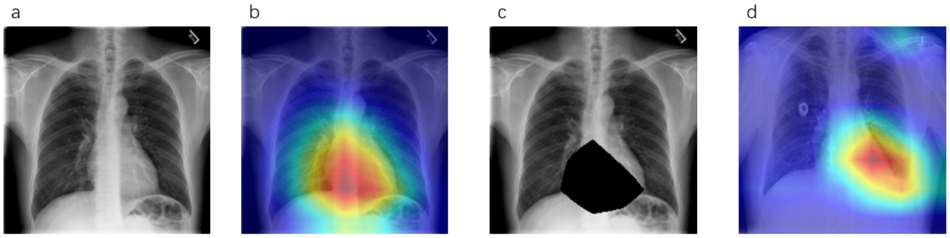


Figure 13: From left to right there are a, b, and c three images. The a image is a sample from train dataset with label "No Findings". The b image is the interpretation result on this image. The c image is the image after applying stronger data augmentation. The d image is the interpretation result on a sample from train dataset with label "Effusion".

Shortcut Dataset experimented in Section 5.2, in NIH CXR dataset, the shortcut learning on age is implicit so that it can only be revealed through numerical metric instead of human observation. To evaluate the performance, we implement the shorT method in Pytorch because authors of shorT method does not provide source code, and get result that is similar to the original paper[1]. Follow the framework of shorT method, like Table 3, we use the MAE loss of age after transfer learning on age to measure the shortcut learning of age since it represents the degree that feature extractor encode the age feature. Because the shortcut learning of age appears when performing the classification tasks on "effusion" and "no effusion"[1], we retrieve all images with label "No Findings", the total number of which is 60361, and all images with label "Effusion", the total number of which is 13317. We divide the train dataset and test dataset by randomly sampling 10000 images from the "No Findings" class and 2000 images from the "Effusion" class. Then, we adopt the same settings as previous experiments to train the model. The model is successfully trained and converge, but we find unexpected phenomenon in while doing the interpretation, which is shown in the Figure 13. In the image b of Figure 13, the highlight of heatmap focuses on the area between the left and right lungs, but in this area, there is not many information about the age so that the stronger data augmentation in the image c of Figure 13 does not have enough affect on the age information. In comparison, the image d of Figure 13 is interpretation on an image sampled from train dataset with label "Effusion". The highlight of the heatmap focuses on the effusion feature of the lung, but it also cannot reveal the implicit age attribute. We believe that is because Grad-CAM is just an approximation of interpretation of prediction so that some tiny features for prediction will be dominated by the primary features for prediction[14]. And as we expected, the applying of IBDAM has no effectiveness in mitigating the shortcut learning on age, which is shown in Table 3. We also experiment all other similar methods like Eigen-CAM and LayerCAM[11 5], but their interpretation results are more or less the same under this senario.

After finding and understanding the limitation of IBDAM, we immediately go to develop other techniques like activation and gradient guided dropout[6], which can perform augmentation on high-level feature dimension, to improve the performance on the NIH CXR dataset. We have performed some training, but due to time and energy constraints, we cannot make the new technique mature and effective when this report is finished. We will attach the result in the submitted code for your reference.

Table 3: Mean Absolute Loss of Age Prediction after Transfer Learning on Age (-0.00x scale refers to the scale of age learning in shorT method)

Model	Original Model	-0.001 scale	-0.003 scale	-0.005 scale	IBDAM-Cutout
MAE	11.91	14.48	14.87	15.34	**12.03**

6 Conclusion

In this report, we review the shortcut learning problem, its affect to the performance of neural network, and previous methods for solving this problem. Then, based on the understanding of previous methods, we propose a much simpler and more intuitive interpretation-based data augmentation method for mitigating the shortcut learning problem. We reveal the underlying assumption and intuition of IBDAM through MNIST dataset, and evaluate the performance of IBDAM through CIFAR10 dataset. In addition, we further extend the IBDAM to NIH CXR dataset and analyze its limitation. Based on above experiment, there are two direction for future work. The first direction is to design more suitable data augmentation for IBDAM and evaluate it in other real-world shortcut dataset. The second direction is to develop methods that can modify high-level feature of images so that IBDAM can also perform well on implicit features.

References

[1] Brown Alexander, Tomasev Nenad, Freyberg Jan, Liu Yuan, Karthikesalingam Alan, and Schrouff Jessica. Detecting shortcut learning for fair medical ai using shortcut testing. *Nature Communications*, 14(1):4314–4314, 2023. doi: https://doi.org/10.1038/s41467-023-39902-7.

[2] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[3] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, Nov 2020. doi: 10.1038/s42256-020-00257-z.

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

[5] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021.

[6] Rohit Keshari, Richa Singh, and Mayank Vatsa. Guided dropout. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4065–4072, 2019.

[7] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

[8] Xu Luo, Longhui Wei, Liangjian Wen, Jinrong Yang, Lingxi Xie, Zenglin Xu, and Qi Tian. Rectifying the shortcut learning of background for few-shot learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 13073–13085. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/6cfe0e6127fa25df2a0ef2ae1067d915-Paper.pdf`.

[9] Gaurav Malhotra and Jeffrey Bowers. What a difference a pixel makes: An empirical examination of features used by CNNs for categorisation, 2019. URL `https://openreview.net/forum?id=ByePUo05K7`.

[10] Matthias Minderer, Olivier Bachem, Neil Houlsby, and Michael Tschannen. Automatic shortcut removal for self-supervised representation learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6927–6937. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/minderer20a.html`.

[11] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2020.

[12] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models, 2018.

[13] Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4974–4986. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/27934a1f19d678a1377c257b9a780e80-Paper.pdf`.

[14] 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 (ICCV)*, Oct 2017.

[15] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), Jul 2019. doi: 10.1186/s40537-019-0197-0.

[16] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[17] Hao Zhao, Jikai Wang, Zonghai Chen, Shiqi Lin, Peng Bao, and Meng Xu. Interpretability-mask: a label-preserving data augmentation scheme for better classification. *Signal, Image and Video Processing*, 17(6):2799–2808, 2023.

[18] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.