Explaining the Effects of Data Augmentation Techniques with Convolutional Neural Networks

(Team ID: 03, Team Name: dhz)

Xiongjie Dai
(xdai12)

Yu He
(yuhe6)

Mengjia Zeng
(mengjia6)

***Abstract*—In the application of image classification, one drawback may be the limited access to large data. In this study, we explored one possible solution to the limited amount of data, the data augmentation technique. Some previous studies have reported the effectiveness of traditional data augmentation techniques in improving the performance of image classifier. We used a less-used dataset named as Oxford-IIIT Pet dataset and artificially combined the images into two classes - dogs and cats. By training with three different Convolutional Neural Network (CNN) models with three different seeds after data augmentation, we found that among the five traditional data augmentation techniques, filpping, rotating, cropping, coloring and blurring, flipping would increase the performance of a image classifier under most of the situations. With the simple CNN models, rotating would also perform well, while cropping and coloring did not help with the performance as compared with the dataset without augmentation. Finally, we also discussed some limitations and future work of this study.**

***Index Terms*—Image Classification, Data Augmentation, Convolutional Neural Network**

I. Introduction

In the field of computer vision, deep convolutional neural networks are capable of classifying images more accurately than humans. It is common knowledge that when there is a larger amount of data, an machine learning algorithm will perform better [8]. However, the main drawback of using large amount of data may be the requirement of big data to avoid overfitting. Many application domains of image classification may not have access to big amount of data, and therefore, limit the application of convolutional neural network in the reality.

In order to overcome this drawback, comparing the performances of different data augmentation techniques and choosing the one that improves the performance of image classifiers greatly over various situations may be meaningful and convenient for researchers to classify images based on limited data in the future [7].

Based on the intuition as mentioned above, this study focuses on testing and comparing the improvement of different data augmentation techniques in the field of image classification. By putting the images from the training dataset into the network with the tensor format, the expected outcome should be the updated weights after several epochs with the augmented data. And also, by using the updated weights with the images from the testing dataset, corresponding accuracy with a specific data augmentation technique could be obtained. By comparing the testing accuracy with different augmentation and the one without any data augmentation, the data augmentation technique that performs the best with this convolutional neural network could be selected. We could also know whether a specific type of traditional data augmentation techniques improves the performance of image classifier as compared with those without data augmentation. We would get plots of training and testing accuracy, as well as the training loss from the model. After trying some different models and different seeds, we would draw a conclusion of the effects of different traditional data augmentation techniques used in improving image classification and which data augmentation technique tends to be the best one under various situations.

II. Related Work

As mentioned above, Data Augmentation (DA) has been a great approach to improve the performance of image classifiers. Traditional data augmentation techniques based on affine image transformation or color modification have widely been proved to be a fast and easy way to implement, while also performing well in increasing the size of training dataset and improving accuracy of the final model [4].

Some researchers have explored in the comparison among the effectiveness of different data augmentations to improve the performance of image classifiers. In general, data augmentation has been proved to be a great approach in improving the accuracy of predicting unseen data. Some traditional approaches, such as shifting, zooming, rotating and flipping, on average, could work well in increasing the accuracy in predicting classes of images from 0.855 to 0.890 when comparing between Dogs and Goldfishes, and increase the accuracy from 0.705 to 0.755 when comparing between Dogs and Cats [8]. Another study was conducted by comparing the performance of image classifiers with data augmentation techniques such as GANs, WGANs, fipping, cropping, shifting, PCA jittering and so on, as well as some combinations, on the CIFAR-10 and ImageNet datasets with 10 categories ranging across 200 samples per class, 1,000 samples per class and 5,000 samples per class. It has been found that all the data augmentation techniques could improve the accuracy of specific algorithms. Among traditional data augmentation techniques, with one data augmentation technique, cropping, flipping and rotating performed better than the others, while most of the combinations of two techniques might further improve the performance [4]. However, a different study suggested that data augmentation techniques used to alter geometry of images may not always perform better than those with lighting and coloring [2], while the other study also indicated that another technique, which was random erasing, would perform the best [5].

Based on the previous research, it is still not clear which techniques will perform the best in increasing the size of training dataset and improving the performance of the image classification algorithms. Furthermore, the generally used datasets are always CIFAR-10 and ImageNet, which also limits the application of those studies when using other different datasets. Therefore, our study followed the procedures similar as the studies as mentioned above. We tried a different dataset that was rarely used, other than CIFAR-10 and ImageNet, for this study, and explored among five different traditional data augmentation techniques, including flipping, rotating, coloring, cropping and blurring, before feeding the training dataset into the same model. After the augmentation techniques, it would be easy to find which of the five different techniques works the best with the same model as the image classifier. Furthermore, we tried three different seeds and three different model structures to see whether the best technique would be different, which was a great difference from the study done before. This generalization procedure would help us to find out whether a specific data augmentation techniques would perform the best, or which data augmentation technique has greater probability to perform the best under various situations.

III. Data

We decided to use the Oxford-IIIT Pets dataset [10] consisting of 37 categories of dog breeds and cat breeds.

For simplicity, we artificially combined all the categories of cat breeds (2,400 images) as the class of Cats and twelve randomly selected categories out of twenty-five of dog breeds (about 2,400 images) as the class of Dogs [10]. Therefore, there were two classes in our data, the class of Dogs and the class of Cats, with around 2,400 images per class and 4,800 images in total. The total number of downloaded images of cats is different from the statistics of the original dataset, which is 2,400 rather than 2,371.

We split the whole dataset into training dataset and testing dataset with a split of 0.8 and 0.2. The training dataset included 3,840 images and the testing dataset included 960 images.

We have conducted two different procedures to pre-process our input image data. First, we used the resizing method to convert all the images to a uniform size. The whole dataset consisted of 4,800 images with different resolutions and we converted the resolution of image data to be the size of 256x256, with 3 color channels. Second, we used the transformation method to convert the images into the format of tensors.

After the pre-process, the training dataset was artificially augmented with five different data augmentation techniques, including flipping, rotating, coloring, cropping and gaussian blurring. Then the size of the training dataset increases from 3,840 to 7,680. As for the testing dataset, after preprocessing, none of the data augmentation was used. After data augmentation, the training dataset was loaded with a batch size of 8 and shuffled. The testing dataset was loaded with a batch size of 8, without any shuffling.

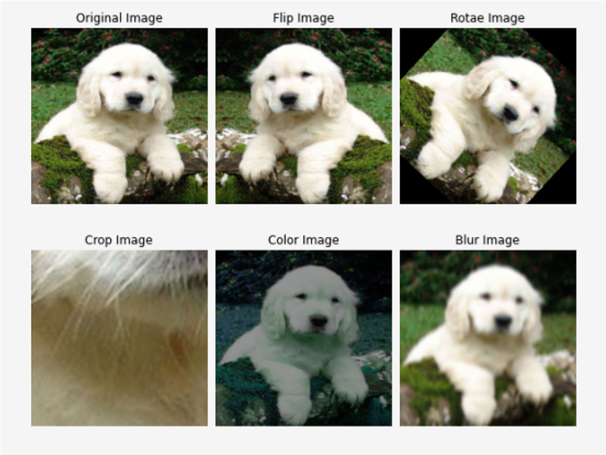


Fig. 1. Example Image with Five Data Augmentation Techniques.

IV. Preliminary Technical Details

We have tried three different small CNN models to classify the images, which we denoted as SmallNet1, SmallNet2 and SmallNet3, and we have also tried three different seeds for each model, which are 1, 2 and 3.

SmallNet1 has two convolutional layers paired with a batch normalization and maxpooling followed by two different convolutional layers. The output is a vector with the weights for each class. The layers of the network are detailed in Fig.2 ReLU is primarily used as the non-linear activation between layers. Moreover, three different regularization techniques have been applied to the SmallNet1: 1D Dropout in the fully Connected layer, 2D Spatial Dropout, L1 and L2 penalty.

SmallNet2 has three convolutional layers paired with a batch normalization following a second layer and maxpooling followed by two different convolutional layers. The output is a vector with the weights for each class. The layers of the network are detailed in Fig.3 ReLU is primarily used as the non-linear activation between layers. Moreover, three different regularization techniques have been applied to the SmallNet1: 1D Dropout in the fully Connected layer, 2D Spatial Dropout, L1 and L2 penalty.

SmallNet3 has three convolutional layers and a larger convolutional layer named as inception before the flatten procedures. The inception layer is divided into four branches, which are processed by convolution with different scales and then the third-dimensional parallel connection of the output results of these four parts is used as the final output result. The output is a vector with the weights for each class. The layers of the network are detailed in Fig.4 ReLU is primarily used as the non-linear activation between layers. Moreover, two

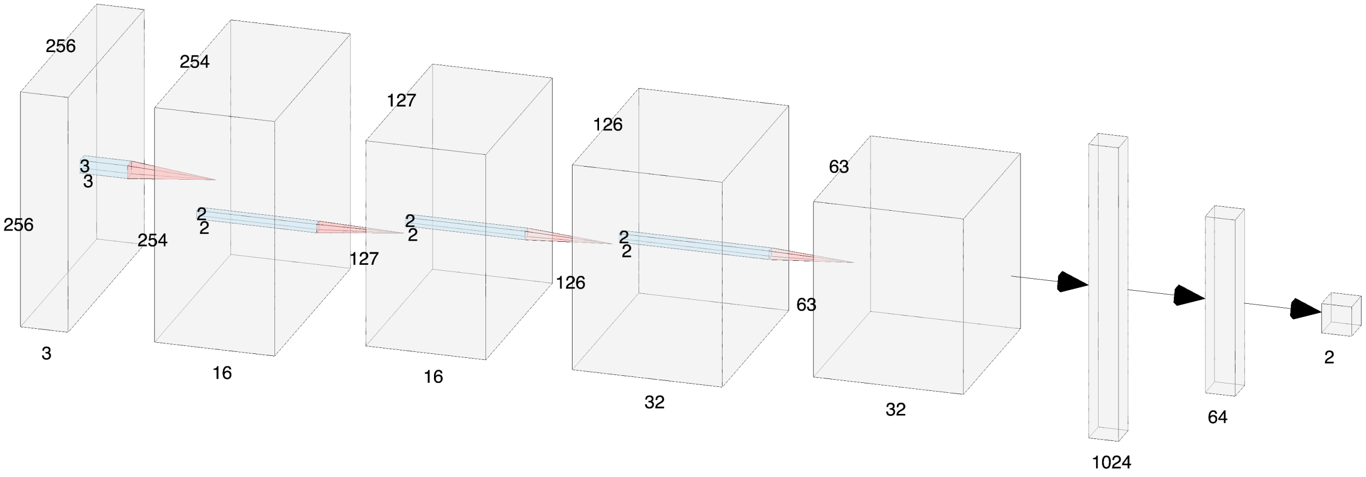


Fig. 2. Diagram of SmallNet1.

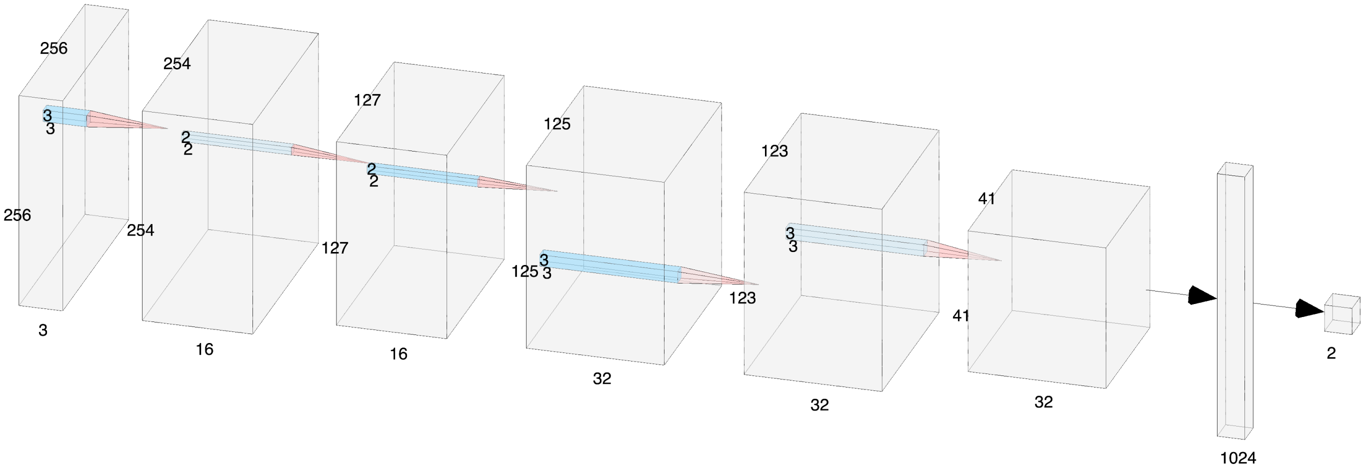


Fig. 3. Diagram of SmallNet2.

different regularization techniques have been applied to the SmallNet1: 1D Dropout in the fully Connected layer and 2D Spatial Dropout.

Since there are only two classes in our dataset, we used the binary cross entropy to be the cost function. And we used the Stochastic Gradient Descent (SGD) as the final optimizer, like the research done by many researchers [3] [9], for SmallNet1 and SmallNet2 with learning rate to be 0.01 and momentum to be 0.8. SmallNet1 and SmallNet2 were conducted with the L2 regularization and the hyper-parameters of weight decay were chosen for every data augmentation under one model in order to minimize the gap between the training and testing accuracy under most of the situations. Among the three different seeds with the same model and data augmentation techniques, the hyper-parameters are kept as the same.

As for SmallNet3 with inception layer, it was conducted without regularization. The same optimization method, SGD, was used with the learning rate to be 0.001 and momentum as 0.8. With this model, no parameter with weight decay was used.

By using different model structures and different seeds, the generalization ability for each data augmentation techniques which could improve the testing accuracy the most as compared with the original dataset could be reported and which data augmentation technique perform the best over different situations would be recommended by this study. The best model with the corresponding seed was selected to further develop the Gradio Application based on the comparison among testing accuracy. The higher testing accuracy meant that the model performs well in predicting the labels of unseen images.

V. Results

After training SmallNet1 with the data augmented with five different data augmentation techniques under the situation with different seeds, we look further into the quantitative results of

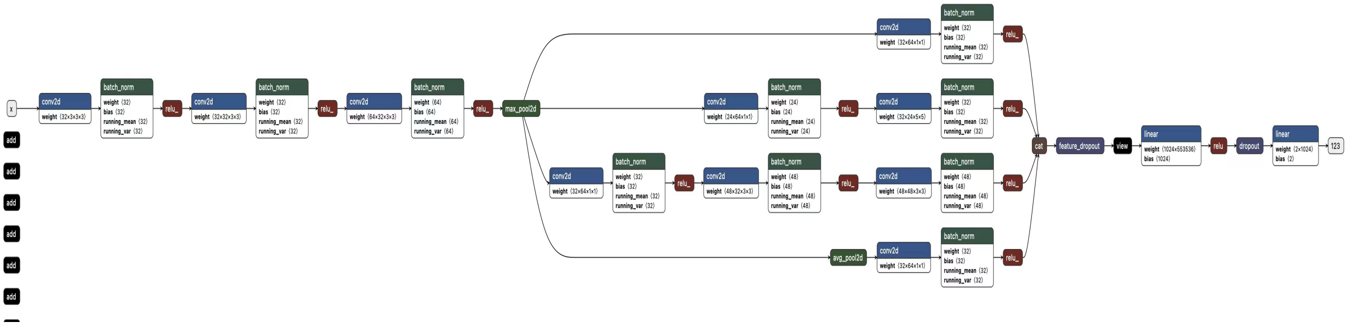


Fig. 4. Diagram of SmallNet3.

the training and testing accuracy after 10 epochs, which is tabulated as below.

Table 1. Accuracy with SmallNet1

Acc.	Flip	Rotate	Crop	Color	Blur	Ori.
Tra.-1	0.855	0.824	0.737	0.921	0.885	0.925
Val.-1	0.620	**0.672**	0.602	0.593	0.605	0.601
Tra.-2	0.626	0.779	0.569	0.500	0.758	0.918
Val.-2	0.616	**0.694**	0.589	0.500	0.618	0.591
Tra.-3	0.885	0.809	0.533	0.836	0.721	0.659
Val.-3	0.658	**0.686**	0.521	0.592	0.594	0.610

As shown in Table 1, with the seed to be 1, four data augmentations, except coloring, perform better than the original data without any augmentation. It is notable that flipping and rotating perform much better than the other two data augmentation techniques, since the other two only improve a little bit as compared with the original one. Rotating performs the best under this situation with a testing accuracy of 0.672 as compared with 0.601 from the original data.

With the seed to be 2, three data augmentations, except coloring and cropping, perform better than the original data without any augmentation. Still, rotating performs the best under this situation with a testing accuracy of 0.694 as compared with 0.591 from the original data.

With the seed to be 3, only flipping and rotating perform better than the original data without any augmentation, both of which increase testing accuracy for higher than 0.04. Rotating performs the best under this situation with a testing accuracy of 0.686 as compared with 0.610 from the original data.

In sum, with SmallNet1, two data augmentation techniques, flipping and rotating, always perform well in increasing testing accuracy as compared with the original data without any data augmentation regardless of the initial weights and seeds. Rotating always perform the best among the five different data augmentation techniques in increasing testing accuracy.

We then train SmallNet2 with the same procedure as that with SmallNet1. The quantitative data of the training and testing accuracy after 10 epochs is tabulated as below.

Table 2. Accuracy with SmallNet2

Acc.	Flip	Rotate	Crop	Color	Blur	Ori.
Tra.-1	0.500	0.865	0.768	0.500	0.675	0.851
Val.-1	0.500	**0.705**	0.641	0.500	0.610	0.664
Tra.-2	0.902	0.859	0.923	0.852	0.900	0.904
Val.-2	**0.748**	0.727	0.639	0.585	0.592	0.706
Tra.-3	0.868	0.850	0.902	0.891	0.927	0.871
Val.-3	0.701	**0.729**	0.635	0.623	0.643	0.655

As shown in Table 2, with the seed to be 1, only rotating performs better than the original data without any data augmentation techniques, increasing the testing accuracy from 0.664 to 0.705.

With the seed to be 2, flipping and rotating perform better than the original data without any augmentation. And flipping performs the best under this situation with a testing accuracy of 0.748 as compared with 0.706 from the original data.

With the seed to be 3, still, only flipping and rotating perform better than the original data without any augmentation, both of which increase testing accuracy for higher than 0.04. Rotating performs the best under this situation with a testing accuracy of 0.729 as compared with 0.655 from the original data.

Therefore, with SmallNet2, rotating always performs well in increasing testing accuracy as compared with the original data regardless of the initial weights and seeds. However, flipping also performs well in some situation with the seed to be 2 or 3, however, when the seed is set to be 1, flipping could not perform well in increasing testing accuracy and the small CNN model fails by stuck at the training or testing accuracy of 0.5.

Last, we then try to train with the SmallNet3 and the quantitative data is tabulated as below.

Table 3. Accuracy with SmallNet3

Acc.	Flip	Rotate	Crop	Color	Blur	Ori.
Tra.-1	0.814	0.666	0.541	0.798	0.833	0.733
Val.-1	**0.705**	0.634	0.534	0.680	0.628	0.646
Tra.-2	0.814	0.664	0.567	0.846	0.881	0.735
Val.-2	0.680	0.605	0.543	**0.692**	0.658	0.631
Tra.-3	0.823	0.695	0.551	0.712	0.846	0.670
Val.-3	**0.723**	0.648	0.565	0.646	0.708	0.596

As shown in Table 3, with the seed to be 1, only flipping and coloring perform better than the original data without any data augmentation techniques. Flipping is still the best data augmentation technique that increases the testing accuracy from 0.646 to 0.705.

With the seed to be 2, flipping, coloring and blurring perform better than the original data without any augmentation. Blurring only improves the testing accuracy for about 0.02, while the other two techniques improve the testing accuracy for about 0.05. And coloring performs the best under this situation with a testing accuracy of 0.692 as compared with 0.631 from the original data.

With the seed to be 3, four data augmentation techniques, except cropping, perform better than the original data without any augmentation. Coloring and rotating improve the testing accuracy for about 0.05, while flipping and blurring increase the testing accuracy for higher than 0.1. Flipping performs the best under this situation with a testing accuracy of 0.723 as compared with 0.596 from the original data.

Therefore, with SmallNet3, flipping always perform well in increasing testing accuracy as compared with the original data regardless of the initial weights and it also performs the best among the five different data augmentation techniques with the seed to be 1 and 3. And coloring always perform well with SmallNet3 with different seeds and perform the best with the seed to be 2. With this SmallNet3, coloring and blurring become efficient in improving testing accuracy, while rotating loses its ability in improving testing accuracy, which may result from the inception layer with convolutional blocks in SmallNet3.

We also draw the accuracy graphs with training and testing accuracy under different situations as the following figure suggests.

From the figure above, the best CNN model trained with the highest testing accuracy should be the SmallNet2 with the seed to be 2 under the data augmentation technique of flipping. The trained CNN model could achieve in predicting between the category of cats and dogs with the accuracy to be 0.748. And we have designed a Gradio application based on this pretrained model, as the Fig.6 shows.

For every image as the input, after the preprocessing, the image would be converted into a format of tensor before feeding into the pre-trained model. After the processing with the model and a procedure using the Softmax function, the final output would be the corresponding probability for the class of Cats and the class of Dogs, just like the work done through the Gradio Application.

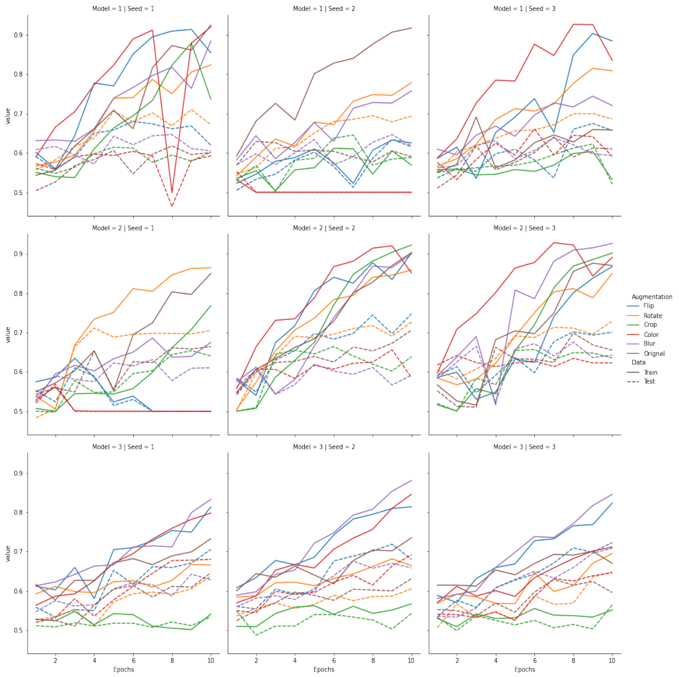
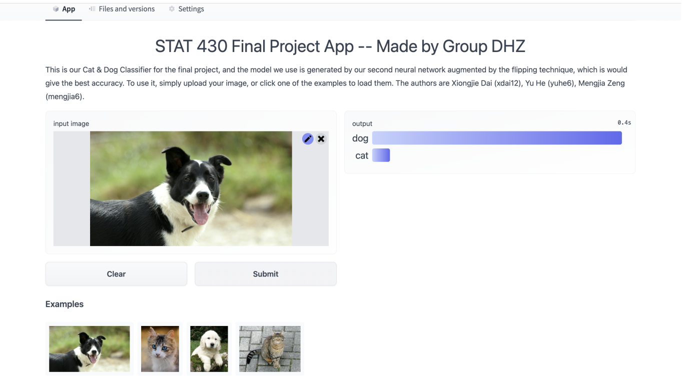


Fig. 5. Training and Validation Accuracy VS. Epochs



STAT 430 Final Project App -- Made by Group DHZ

This is our Cat & Dog Classifier for the final project, and the model we use is generated by our second neural network augmented by the flipping technique, which is would give the best accuracy. To use it, simply upload your image, or click one of the examples to load them. The authors are Xiongjie Dai (xdai12), Yu He (yuhe6), Mengjia Zeng (mengjia6).

Fig. 6. Screenshot of the Built Gradio Application.

VI. Conclusion and Future Work

We notice that the improvement of testing accuracy differs from different CNN models and different seeds.

However, we could still draw some conclusions. First, for similar CNN models without the inception layer including some convolutional blocks, flipping and rotating always perform better than the other three data augmentation techniques. Based on the results with several different seeds, this study suggests that for simple CNN models with several convolutional layers, flipping and rotating, especially rotating, could help with improving the accuracy of a image classifier. On the other side, cropping, coloring and blurring may not be useful.

Second, for the CNN model with convolutional blocks, the SmallNet3, flipping still performs well. The effect of coloring in improving the predicting accuracy has also improved a lot under this situation. For some more complex CNN models, it will be efficient to start from flipping and coloring rather than the other traditional techniques as mentioned in this article.

This study also has several limitations. First, the study only records results after 10 epochs, because with more epochs, some models may face the problem of overfitting. In the future, more choices of hyper-parameters could also be tested. Second, flipping always perform well in improving testing accuracy. However, under using SmallNet2 with the seed to be 1, the training accuracy and testing accuracy may be stuck at 0.5, meaning that the choice of seeds may not be good enough under this situation.

As for the future work, first, this study focuses building CNN models classifying images between two classes. And the future work could be done by extending this study into building CNN models and augmenting data from multiple classes rather than that with only two classes. By generalizing the study with more classes, it would also be worthwhile to check whether the predicting accuracy for each different class has all been improved, since one study reported that even though the data augmentation technique would increase the whole testing accuracy, testing accuracy for some specific classes may be sacrificed and decreased greatly [1]. Last, in the future, it would be worthwhile to try more data augmentation techniques, for example, Generative Adversarial Networks (GANs) or some other newly developed techniques, like Sample-Pairing [3], as well as some combination between different data augmentation techniques, and compare with the traditional data augmentation techniques to see which data augmentation technique is more likely to be the best one that improves the accuracy in predicting labels and whether combinations of data augmentation techniques would perform better than a single one.

Our group has learned a lot by doing this research through knowing further about using Pytorch to exert different data augmentation techniques to process a image , write the Convolution Neural Network and use SGD to optimize the training process. We know further about writing articles with IEEE format.

References

[1] Balestriero, Randall, Leon Bottou, and Yann LeCun. ”The Effects of Regularization and Data Augmentation are Class Dependent.” arXiv preprint arXiv:2204.03632 (2022).

[2] Hu, Benlin, et al. ”A preliminary study on data augmentation of deep learning for image classification.” arXiv preprint arXiv:1906.11887 (2019).

[3] Inoue, Hiroshi. ”Data augmentation by pairing samples for images classification.” arXiv preprint arXiv:1801.02929 (2018).

[4] Mikołajczyk, Agnieszka, and Michał Grochowski. ”Data augmentation for improving deep learning in image classification problem.” 2018 international interdisciplinary PhD workshop (IIPhDW). IEEE, 2018.

[5] O'Gara, Sarah, and Kevin McGuinness. ”Comparing data augmentation strategies for deep image classification.” (2019).

[6] Shijie, Jia, et al. ”Research on data augmentation for image classification based on convolution neural networks.” 2017 Chinese automation congress (CAC). IEEE, 2017.

[7] Shorten, Connor, and Taghi M. Khoshgoftaar. ”A survey on image data augmentation for deep learning.” Journal of big data 6.1 (2019): 1-48.

[8] Perez, Luis, and Jason Wang. ”The effectiveness of data augmentation in image classification using deep learning.” arXiv preprint arXiv:1712.04621 (2017).

[9] Zheng, Qinghe, et al. ”A full stage data augmentation method in deep convolutional neural network for natural image classification.” Discrete Dynamics in Nature and Society 2020 (2020).

[10] Original Dataset Link: https://paperswithcode.com/dataset/oxford-iiit-pets Dataset link: https://drive.google.com/drive/folders/1F6gQmV8LCBAoYN5YWf_-KIBWaVvXAIjl?usp=sharing

Code and Application

The code can be accessed through this link https://huggingface.co/spaces/yuhe6/final_project

Contribution

Every member in our group has greatly contributed to our progress and we have equal contributions.
Xiongjie Dai 33 percent Write codes of some models, change the parameter of weight decay and run the whole scripts with HAL. Help with the proposal and final report.
Yu He 33 percent Write codes to process the input data, discuss with the professor to solve problems and write the Gradio Application. Help with the proposal and progress report.
Mengjia Zeng 33 percent Read some papers and raise the ideas. Write the codes of some models, try to change the parameter of weight decay and write the final report and slides.