Data augmentation techniques for low-resolution image classification

COMP 7250 Mini-Project
Adilet Uvaliyev (24451703)

I. Introduction

Image classification is an important task with many useful applications [1]. Deep learning models can achieve a high accuracy rate on this task [2]. However, they require substantial amount of training dataset and might suffer from the issue of overfitting if the dataset size isn't enough. Various regularization techniques exist to address this issue [3]. Data augmentation is one type of such technique and it produces artificial images to increase the dataset size [4]. There are various types of augmentations including traditional ones such as rotation, color change or more advanced ones such as techniques based on GANs [5]. In this project, the effectiveness of geometric and color based augmentations for binary low-resolution image classification task was studied. The results of experiments show that random search is effective for finding optimal parameters and contrast jitter augmentation is most effective for distinguishing automobile and truck images. Also, the effectiveness of augmentation increases as the dataset size becomes lower. Lastly, the experiments show that starting augmentation at intermediate stages of training gives better performance than starting from the beginning.

The rest of the report is structured as follows. The problem definition is summarized in Section 2. The methodology including dataset description, pre-processing, data augmentation techniques and training details is explained in Section 3. The experimental results are included in Section 4. Finally, Section 5 contains the conclusion.

II. Problem Definition

The main problems addressed in this project can be summarized as follows:

- **Perform random search hyperparameter tuning to find optimal parameters for data augmentation techniques:** This project evaluates the effect of hyperparameter tuning using random search for data augmentation effectiveness.
- **Compare different types of data augmentation techniques for binary low-resolution image classification:** This project experiments with 3 geometric and 3 color based data augmentation techniques for images having small resolution.
- **Evalute the data augmentation effectiveness in terms of dataset size:** In this project 5 different dataset sizes are used to test the data augmentation usefulness.
- **Evaluate the effectiveness of injecting data augmentation at intermediate epochs during training:** In this project, data augmentation was injected at different epochs and the results were compared.

III. Methodology

A. Dataset Description

CIFAR-10 dataset was used in this project [6]. This dataset contains 600000, 32 by 32 color images with 10 classes. Automobile and truck classes are used in this project. To avoid a class imbalance problem, the classes have an equal number of images. For the main experiment, 700 images in total is used and for later experiments, 429, 572, 857, and 1000 images were used. The dataset is divided into train and validation sets in roughly 70/30 ratio. The purpose of using only two classes and a smaller number of images is to reduce the training time and produce the need for using regularization techniques.

B. Preprocessing

The training and validation set were pre-processed by resizing into 224 by 224 images, conversion into a tensor, and normalizing into a mean of [0.485, 0.456 0.406] and standard deviation of [0.229,0.224,0.225]. The validation set didn't have further preprocessing and the training set was further preprocessed by data augmentation techniques described in part C.

C. Data augmentation techniques

In this project, 3 types of geometric and 3 types of color based data augmentation techniques were selected. For the convenience, all hyperparameters were defined in terms of a parameter alpha. This parameter ranges from 0 to 1 and is drawn randomly from a uniform distribution. Here is a description of augmentation methods used in the project:

Horizontal Flip: Randomly flips the image horizontally with a given probability alpha. This value ranges from 0 to 1.

Vertical Flip: Randomly flips the image vertically with a given probability alpha. This value ranges from 0 to 1.

Rotation: Rotates all images with the angle of rotation drawn from a uniform distribution from a user specified -angle to +angle. In this project, the angle is defined as 360*alpha, where alpha ranges from 0 to 1.

Hue Jitter: Randomly changes the hue of an image with a factor drawn from a uniform distribution between -alpha / 2 and + alpha / 2. Hue refers to the dominant color in an image.

Contrast Jitter: Randomly changes the contrast of an image with a factor drawn from a uniform distribution between 1-alpha and 1 + alpha. The contrast means the difference between the brightest and darkest pixels in an image.

Saturation Jitter: Randomly changes the saturation of an image with a factor drawn from a uniform distribution between 1-alpha and 1 + alpha. The saturation refers to the amount of color in an image.

Baseline: Apart from these, the baseline model was trained without using any augmentation.

D. Training details

Resnet-18 model with pre-trained weights used in this project [7]. Only the final classification layer was learned during training. The model was trained using an Adam optimizer with a 0.0001 learning rate for 10 epochs. The batch size is 8.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Random search hyperparameter tuning

For each data augmentation technique, the model was trained for 5 times. At each time the data augmentation hyperparameter alpha, was drawn randomly from a uniform distribution from 0 to 1. For

TABLE I: Effect of random hyperparameter search on data augmentation (training size = 500 images)

Augmentation	Worst Validation Acc	Best Validation Acc
Horizontal Flip	80.50	86.00
Vertical Flip	77.00	83.00
Rotation	83.50	87.50
Hue Jitter	75.00	83.00
Contrast Jitter	79.00	88.00
Saturation Jitter	81.00	85.50

TABLE II: Effectiveness of different types of data augmentation techniques (training size = 500 images)

Augmentation	Training Acc	Validation Acc
Horizontal Flip	**82.60**	86.00
Vertical Flip	80.60	83.00
Rotation	80.60	87.50
Hue Jitter	81.20	83.00
Contrast Jitter	79.80	**88.00**
Saturation Jitter	79.20	85.50
None	82.20	86.00

each data augmentation technique, the best and worst performance in terms of validation accuracy is shown in Table I. This shows the importance of hyperparameter tuning as the gap between best and worst is considerable. The results of all experiments are shown on Tables V, VI. The random search hyperparameter was selected instead of the grid search because it tends to produce better hyperparameters. [3].

B. Effectiveness of different types of augmentations

Table II shows the comparison between baseline technique and augmentation techniques with the best hyperparameters. The contrast is the best augmentation technique in terms of validation accuracy, outperforming the baseline by 2%. This might be because under low image resolution contrast change might resemble most the original images. Thus, it can increase the dataset size without introducing many non-realistic artificial features. However, this technique didn't improve the training performance. This shows that data augmentation techniques in this task improve the generalization performance and reduce overfitting. It is better to use data augmentation even if training accuracy is lower than the baseline model.

C. Dataset size impact on data augmentation effectiveness

In order to evaluate the impact of dataset size on the data augmentation performance, the baseline model and model with best performing augmentation was tested at training size with 300,400,500,600,700 images. The result is shown in table III. The augmentation technique had the highest performance at 400 images training size. The accuracy gain is negative when training with 300 images. This might be because the data augmentation impact is too much when the training set size is small and model can't learn well the original data distribution. As expected, accuracy gain is positive but less with bigger size dataset, showing less need for regularization techniques as the dataset becomes enough for the given task. The figures 1 and 2, show the comparison of accuracy and loss curve for the baseline model and the model with best augmentation. The curves show that using augmentation consistently improves performance in terms of both training and evaluation.

TABLE III: Data augmentation effectiveness in terms of dataset size

Train size	Val size	Baseline Val Acc	Augmentation Val Acc	Acc gain
300	129	78.91	78.13	-0.78
400	171	78.82	84.12	**5.30**
500	200	86.00	88.00	2.00
600	257	83.20	84.38	1.18
700	300	85.33	86.67	1.34

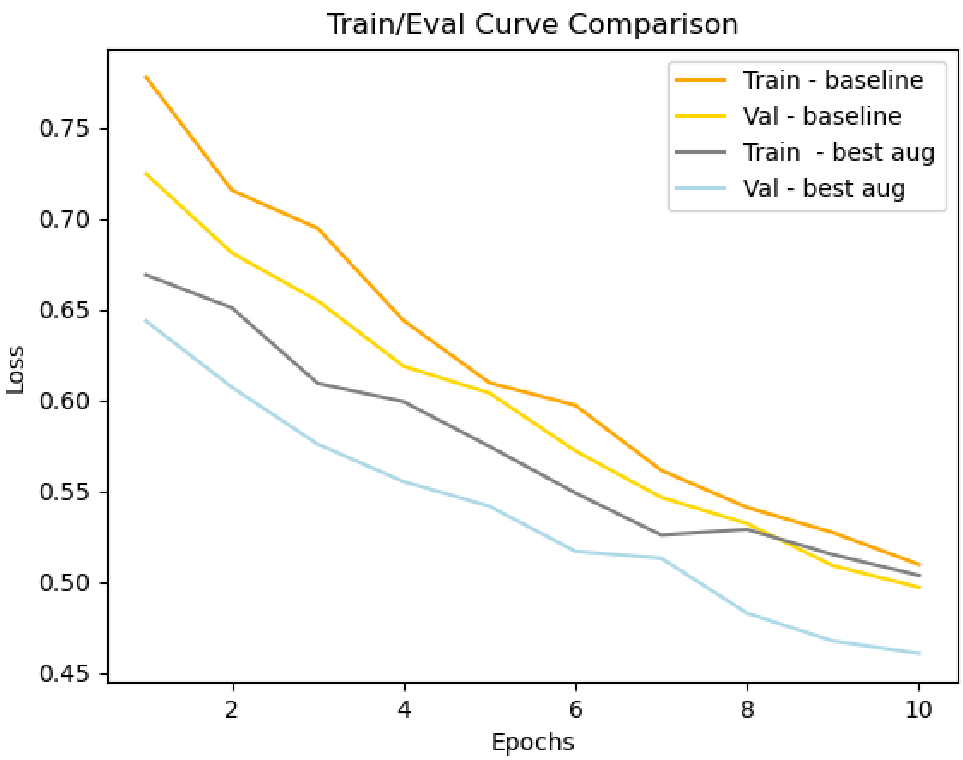

Fig. 1: Training and Validation Loss Curve (training size = 400)

D. Injecting data augmentation starting at different epochs

According to a previous study, data augmentation has shown better performance when starting injecting augmentations at intermediate epochs [8]. In this project, the augmentation technique was injected starting at epochs at 1,3,5,7,9, and the results is shown in table IV. Injecting at epoch 5, demonstrated the best performance, gaining 0.58 % evaluation accuracy compared to augmentation starting from the beginning. This could be because the overfitting starts at intermediate epochs and there is no need to start augmentation from the first epoch. At initial epochs, it is better for a model to learn from original images and artificial features of augmentation might have a negative impact.

V. Conclusion

In this project, traditional augmentation techniques were applied for binary image classification and their effectiveness in terms of type, dataset size, and training stage was evaluated. The experimental results show that random search hyperparameter tuning is effective for finding optimal parameter, contrast jitter is most suitable technique, lower dataset size result in better augmentation effectiveness, and

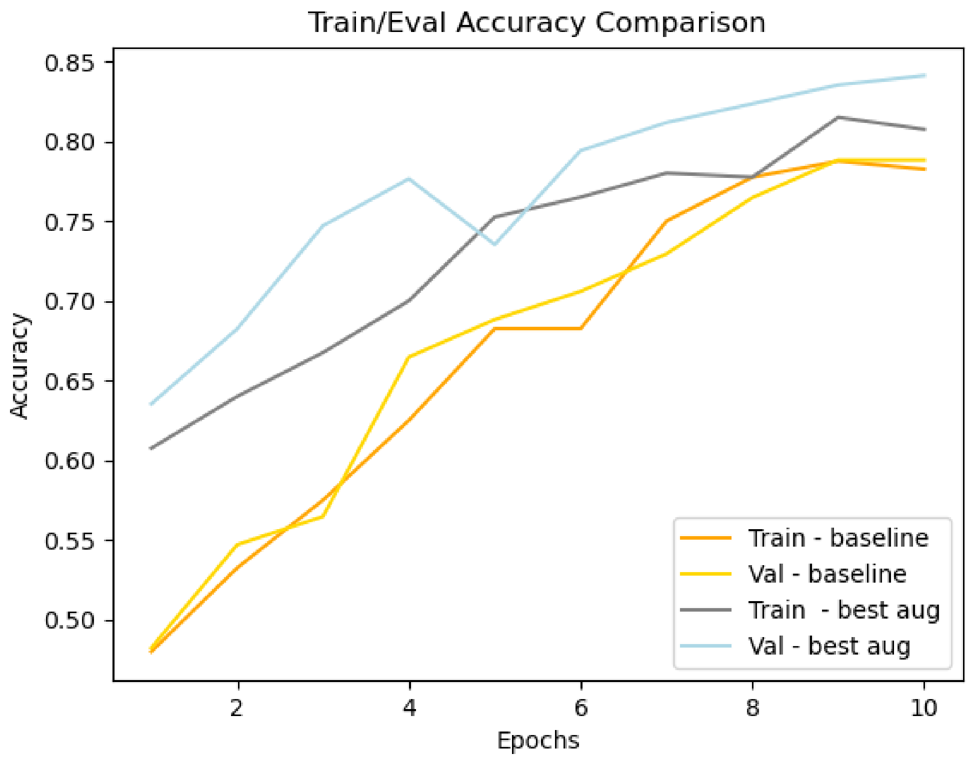

Fig. 2: Training and Validation Acc Curve (training size = 400)

TABLE IV: Data augmentation injection starting at different epochs (training size = 400)

Augmentation	Start Epoch	Train Loss	Val Loss	Train Acc	Val Acc
Contrast Jitter	1	0.4969	0.4663	78.00	85.47
Contrast Jitter	3	0.5442	0.4921	74.50	80.23
Contrast Jitter	5	0.5118	0.4566	79.25	**86.05**
Contrast Jitter	7	0.5405	0.5238	78.50	77.91
Contrast Jitter	9	0.5006	0.4992	81.25	80.81

introducing augmentation at the intermediate stage of training is optimal. Due to the time limitation only 6 augmentation techniques were tested.

REFERENCES

[1] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," *Remote Sensing*, vol. 13, no. 22, p. 4712, 2021.

[2] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[3] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[4] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[5] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, p. 100258, 2022.

[6] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[8] K. McGuinness and S. O'Gara, "Comparing data augmentation strategies for deep image classification," 2019.

TABLE V: Geometric augmentation with random hyperparameters (training size = 500)

Augmentation	Alpha	Training loss	Validation loss	Training acc	Validation acc
Horizontal Flip	0.63	0.4711	0.4621	82.60	86.00
Horizontal Flip	0.09	0.4489	0.4581	83.40	81.00
Horizontal Flip	0.02	0.4458	0.4506	84.60	83.00
Horizontal Flip	0.11	0.4843	0.4784	80.40	80.50
Horizontal Flip	0.84	0.4648	0.4657	82.40	82.50
Vertical Flip	0.02	0.4986	0.4627	80.60	83.00
Vertical Flip	0.22	0.5173	0.4929	77.60	80.00
Vertical Flip	0.14	0.4825	0.4699	81.20	78.50
Vertical Flip	0.33	0.5295	0.4801	76.00	79.50
Vertical Flip	0.58	0.5556	0.5381	76.00	77.00
Rotation	0.58	0.4706	0.4491	82.40	83.50
Rotation	0.65	0.4876	0.4651	83.00	83.50
Rotation	0.01	0.4736	0.4506	82.40	84.50
Rotation	0.17	0.4880	0.4564	80.60	83.50
Rotation	0.68	0.4803	0.4520	80.60	87.50

TABLE VI: Color based augmentations with random hyperparameters (training size = 500)

Augmentation	Alpha	Training loss	Validation loss	Training acc	Validation acc
Hue Jitter	0.78	0.4937	0.4898	79.20	80.50
Hue Jitter	0.56	0.5021	0.5037	76.60	75.00
Hue Jitter	0.58	0.5181	0.4685	78.80	81.50
Hue Jitter	0.33	0.4770	0.4532	81.20	83.00
Hue Jitter	0.96	0.5123	0.4901	79.00	81.50
Contrast Jitter	0.46	0.5193	0.5142	78.40	79.50
Contrast Jitter	0.58	0.4916	0.4540	79.80	88.00
Contrast Jitter	0.53	0.4807	0.4719	82.20	82.00
Contrast Jitter	0.50	0.4580	0.4502	82.20	79.00
Contrast Jitter	0.22	0.4550	0.4242	82.20	86.50
Saturation Jitter	0.30	0.4827	0.4658	81.40	81.50
Saturation Jitter	0.27	0.4720	0.4803	80.80	81.00
Saturation Jitter	0.08	0.4685	0.4616	84.00	80.50
Saturation Jitter	0.30	0.4685	0.4732	84.20	82.00
Saturation Jitter	0.45	0.4891	0.4554	79.20	85.50