

Study of Data Enhancement Techniques for Steel Surface Defect Detection

Zhiquan Sun

*School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China
szq2022112532@my.swjtu.edu.cn*

Abstract: The steel surface defect detection is always worth exploring; for this, Convolutional Networks for Biomedical Image Segmentation (U-Net) is one of the commonly used techniques for solving such problems, which is often used with Residual Network (ResNet) or other encoders. However, apart from the main technique structure, data augment methods for segmentation problems have not been widely and deeply studied, which is the focus of this paper. The purpose of this research is to investigate the impact of different kinds of data augmentation methods on the model training effect. Specifically, this study mainly trained the model by applying different kinds of data augmentation methods separately and comparing them through the set metrics: Dice coefficient and Intersection over Union (IoU). Different kinds of data augmentation methods were grouped according to their functions. After obtaining the eventual data of each group, the good-performing ones were selected for further combining or adjusting the parameters. This study is conducted on Severstal: Steel Defect Detection from the Kaggle platform. The final results show that data augment methods do not necessarily lead to a huge improvement in the metrics with the used data set and the framework. The research also analyzes the potential reasons for the result. Moreover, the prevention of overfitting by these methods is still not negligible. The study has value for the use of data augment methods, especially in the treatment of segmentation problems, and has some implications for more in-depth research.

Keywords: Steel Surface Defect Detection, Segmentation, U-Net, ResNet, Data Augmentation.

1. Introduction

Steel is one of the most important industrial products. Recent researches investigating the impact of Industry 4.0 on supply chain management suggests that it is also one of the key foundations of Industry 4.0 [1]. However, in steel production, there always exist various defects, such as inclusions, patches, and scratches, which are one of the bottlenecks in the industry's growth [2]. Moreover, optimal supply chain configurations rely on energy configurations, the emergence of new steel mills near high-quality renewable energy sources, regional alliances to create green steel markets, and billions of dollars of investment [3]. As a result, there is still a current need to find reasonable ways to deal with this situation.

In recent years, the development of deep learning and computer-vision-related technologies has greatly assisted in the detection of relevant defects, which to some extent has contributed to the quality of steel production. Current deep learning-based automatic detection algorithms have received much

attention, and these algorithms mainly perform feature extraction through Convolutional Neural Networks (CNNs) and then use neural networks to realize classification or segmentation [4].

CNNs have been proven to be an effective way to extract features, providing a new method to address the issue of low accuracy in recognizing surface defects on steel surfaces, while the Convolutional Networks for Biomedical Image Segmentation (U-Net) networks hold great promise for semantic segmentation applications [5]. In former research, with the methods, the existing model could perform well with a Dice coefficient of 0.7893 and an IoU of 0.6453, and some of the better models' Intersection over Union (IoU) can have correlation values of 0.85 or higher [5, 6]. These achievements have made it possible to identify most steel surface defects with a high degree of accuracy, helping quickly recognize and respond to the defects. The application of these methods illustrates the feasibility of the technology. However, for industrial needs, methods are still needed for further enhancement in terms of accuracy.

Data management methods are always applied to data preprocessing, especially in cases where the risk of overfitting is high or there is a need to improve the performance of classification models or adapt to different data distributions. However, it is worth mentioning that data augmentation has not been sufficiently specialized in this topic, which may help in model optimization and further research. By utilizing augmentation techniques, the applicability of Artificial Intelligence (AI) models in tasks involving unbalanced datasets can be significantly improved, which greatly enhances the generalization ability of AI models and increases the robustness of the system [7]. Commonly used data augmentation methods include flipping, rotating, panning, etc. For the steel surfaces, brightness adjustment, contrast adjustment, and other color transformation methods can also be used. By applying such techniques at a deeper level, it is possible to promote relevant developments in modeling techniques.

The research of this paper focuses on the automatic detection and classification of steel surface defects with the help of deep learning methods. Specifically, it is a U-Net-based neural network. In this work, the Residual Neural Network-18 (ResNet-18) is utilized as an encoder to extract high-level features from the input data images, which are then transferred to the decoder part of the U-Net model for segmentation. Next, the model tries to enhance model generalization by equipping data augmentation techniques, including geometric transformation, noise and blur, image deformation, color adjustment, etc., with the support of the albumentations library. To fit the multi-classification problem better, the model uses the Binary Cross Entropy (BCE) loss function. During training, the Dice coefficient and IoU are calculated for validation. The output is a set of segmented masks of defects, which could be converted into pixels on images. The core of this research is to investigate the effect of different data augmentation methods on model training by replacing and combining them with a comparison of the Dice coefficient and IoU.

2. Methodology

2.1. Dataset

The dataset is provided by Severstal: Steel Defect Detection [8] on the Kaggle platform. By March 12, 2025, the dataset contains 12,568 training images with a total of 7,096 labeled defects; each of the images has a size of 1600*256 pixels. This dataset contains four classes of defects, including inclusions, cracks, scratches, and scales, corresponding to Figure 1.

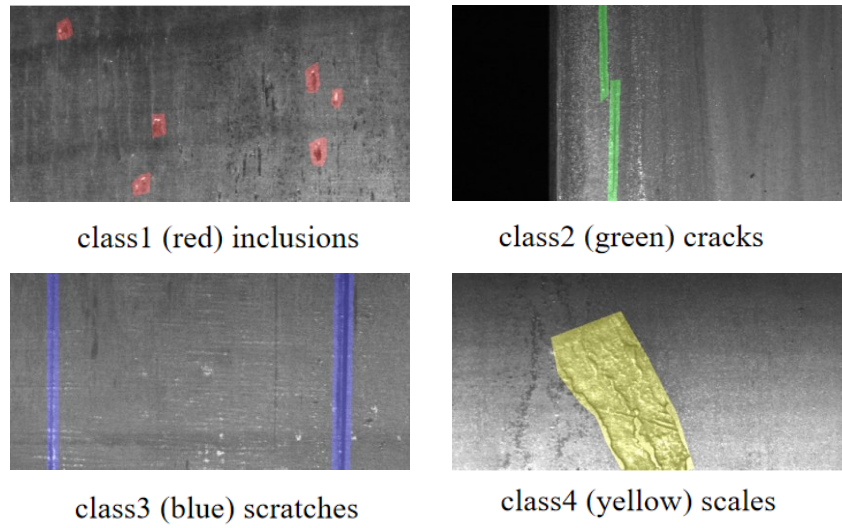


Figure 1: 4 classes of defects in dataset (picture credit: original)

The images are high-resolution industrial-grade Red, Green, Blue (RGB) images with high labeling accuracy, providing a pixel-by-pixel mask of defect locations. The dataset is an authoritative benchmark in the field of industrial defect detection, and its standardized format and multi-scenario coverage provide a reliable basis for model generalization capability validation.

2.2. Proposed approach

As shown in Figure 2, the overall process of this research is as follows: first, the images and corresponding masks are extracted from the dataset. Second, the data augmentation methods supported by the albumentations library are performed, which is the focus of the research. Then the data is converted into a suitable tensor and passed into the U-Net model for processing. After processing by the model, the mask for predicting defects can be obtained.

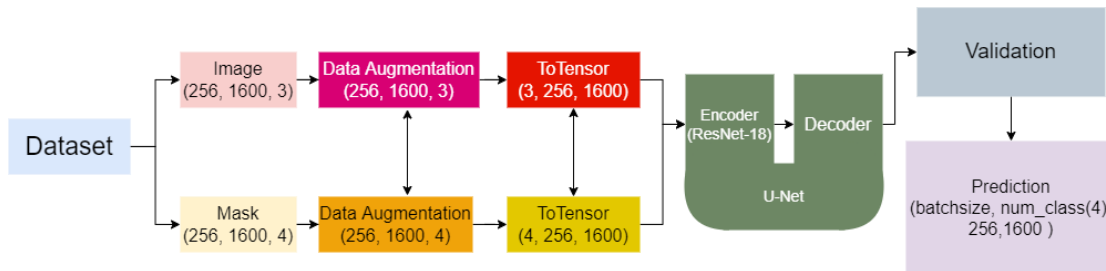


Figure 2: Overall process (picture credit: original)

2.2.1. Model architecture

U-Net was applied in the field of biomedical images initially and achieved great success. Due to its flexibility, there are now many variants of it. In the biomedical field, U-Net is often used for segmentation of medical images, improving the productivity of segmentation of disease-affected areas such as the brain, lungs, retina, and liver [9]. U-Net-based techniques are not only effective in medical images but also valuable in the problem of detecting steel defects where segmentation is required.

As far as the model architecture, U-Net is an encoder-decoder network whose structure can be changed flexibly, which is one of the reasons why the scheme is adopted in this study. As Figure 3

showed, the U-Net network architecture combines shallow and deep features through a U-shaped network with hopping connections to generate semantic segmentation maps. The encoder employs convolution-max pooling for four downsampling operations of the input image to generate four hierarchical feature maps. The decoder fuses its feature maps with those from transposed convolution via skip connections. The final layer predicts the semantic map by calculating loss against ground truth labels.

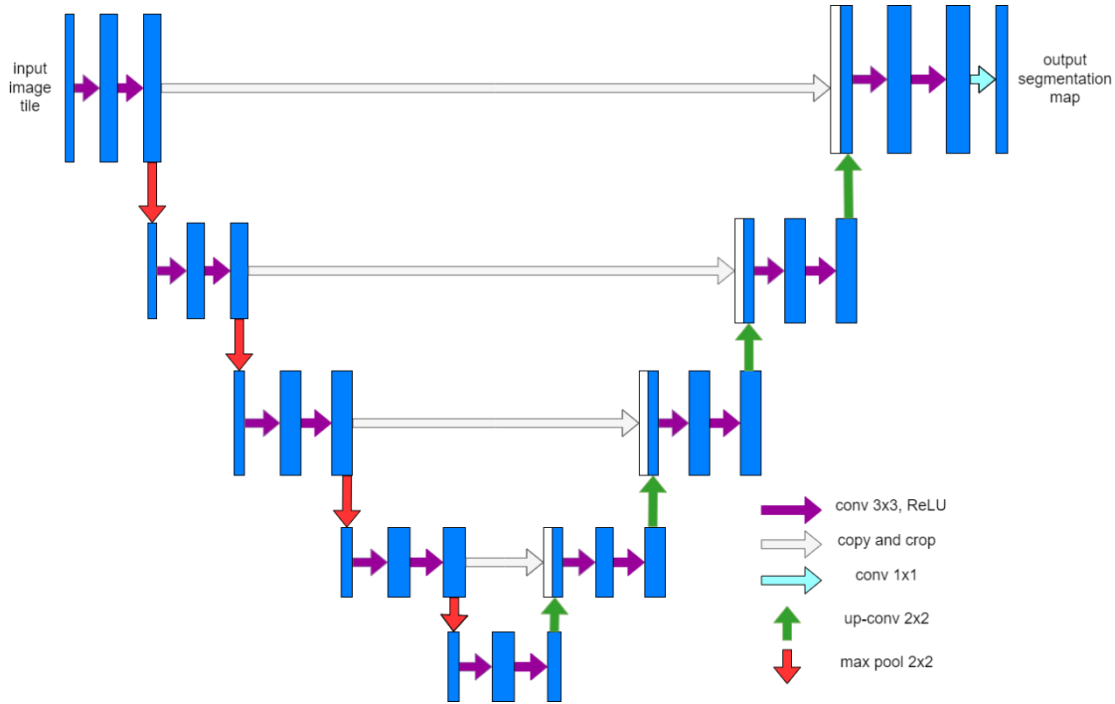


Figure 3: U-Net scheme (picture credit: original)

This structure is simple and efficient, balancing high accuracy with low data dependency in tasks such as medical image segmentation [9, 10]. The same holds also for similar problems of steel defect detection. Overall, the U-Net model is well-suited for the research topic. The model used in this study is based on the traditional U-Net architecture, with modifications solely to the encoder. Although it possibly has certain limitations compared to some of the latest variants, it remains sufficiently compelling for the specific research topic for its base scheme. In the study, the tensor obtained after preprocessing is passed into the U-Net model, and after processing, it gets its output prediction mask.

2.2.2. Residual network (ResNet)

ResNet is a CNN-based architecture that introduces residual blocks and skip connections, enabling the network to learn more abundant and representative features [11]. Residual blocks allow the network to learn residual mappings. By formula presentation, that is $F(x) = H(x) - x$, where $H(x)$ represents the target latent function and x is the input. This architectural design significantly reduces training difficulty by bypassing gradient degradation, allowing the network to efficiently capture hierarchical features ranging from low-level spatial details (e.g., edges, textures) to high-level semantic representations. In this study, ResNet-18 was utilized as the encoder for U-Net. Previous studies have demonstrated the superior performance of ResNet-18 in classification problems in the case of a number of datasets to achieve improved classification accuracy [12, 13]. The ResNet-18 used in this study has a shallower architecture, which can accelerate model training compared to

multi-layer ResNet models with more layers while providing significant convenience for the progress of the research.

2.2.3. Data augmentation

Albumentations is a fast and flexible image augment library designed for computer vision tasks, mainly used in image classification, target detection, semantic segmentation, instance segmentation, and other fields. In the study, data augmentation is realized with the support of this library, and comparative experiments are conducted by replacing different combinations of augmentation. Table 1 shows the classification of different kinds of methods. Each classification is used for different groups of experiments and is validated after 30 epochs of training, when the model training is able to reach a stable level. After each experience of each group, some of the best-performing methods would be used for combination or take hyperparameter adjustments for further study.

Table 1: Method classification

Group Number	Method	Manipulation
1	Baseline	NULL
2	Flip	Horizontal Flip, Vertical Flip
3	Geometric Transformation	Rotate, Shift Scale Rotate
4	Degradation	Gauss Noise, Motion Blur, Median Blur
5	Structural Disruption	Grid Distortion, Elastic Transform, Optical Distortion
6	Color Space Change	Hue Saturation Value, Random Brightness Contrast, Contrast Limited Adaptive Histogram Equalization (CLAHE), Random Gamma
7	Nature Simulation	Random Fog, Random Rain, Random Sun Flare
8	Size Adjustment	Random Crop, Resize

The baseline group applies no augmentation methods to preserve the original data distribution for comparison. The flip group horizontally or vertically flips images with a 50% probability to simulate different viewing angles. The geometric transformation group randomly rotates images by 0-45 degrees and applies affine transformations with up to 6.25% translation and 10% scaling, also with a 50% probability, to mimic object orientation changes. The degradation group randomly (50%) corrupts images with Gaussian noise (variance between 10 and 50), motion blur (levels from 1 to 7), or median blur (from 1 to 5) to simulate poor imaging conditions. The structural disruption group introduces moderate non-linear and affine deformations, grid distortion with a 5-step grid and a 30% limit, or optical lens effects with a 50% chance to challenge structural consistency. The color space change group randomly (50%) adjusts color attributes (hue within ± 20 , saturation within ± 30 , brightness within ± 20) and contrast (using CLAHE with 8x8 tiles and a clip limit of 3.0) to account for lighting variations. The nature simulation group adds environmental effects like fog, rain, or sun flare with a 50% probability to replicate real-world weather conditions. Finally, the size adjustment group uniformly resizes all images to 256x256 and then crops them to 224x224 pixels with 100% execution to standardize input dimensions.

2.2.4. Loss and evaluation metrics

This paper uses the Binary Cross Entropy Loss (BCE Loss) for binary classification problems that calculates the cross entropy between the probability distribution of the model output and the true label. It could be calculated with this formula:

$$BCE\ Loss = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (1)$$

In which y_i represents the true label of the i -th sample, N represents the number of samples, p_i represents the predicted probability of the i -th sample. In the study, the Sigmoid activation function was used by converting the multi-classification problem into multiple binary classification problems, thus p_i above was replaced by $\sigma(z_i)$, where z_i is the logits of the model. That is the BCE With Logits Loss used in the research:

$$BCE\ With\ Logits\ Loss = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))] \quad (2)$$

It combines Sigmoid activation and BCE Loss, which makes the computation more efficient and avoids numerical stability issues (e.g., zero-value problems that can occur when computing logarithms). In the fields of image segmentation and computer vision, the Dice coefficient is widely used to evaluate the degree of overlap between the predicted results and the true labels. The Dice coefficient can be calculated with the formula below:

$$Dice\ coefficient = \frac{2|A \cap B|}{|A| + |B|} \quad (3)$$

In which A represents the prediction result and B represents the true label. The higher the values, the better agreement between the prediction and the true label. The Dice coefficients are simple to calculate, and the numerical interpretation is intuitive, focusing directly on the segmentation quality of the target region. In the topic and research direction of this study, Dice coefficients can also intuitively and sensitively reflect the training of the model. Therefore, it is adopted as one of the evaluation metrics in this study. Second, IoU is another common evaluation metric for computer vision and machine learning, also used to measure the degree of overlap between two regions or sets, as:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (4)$$

Similar to the Dice coefficient, A represents the prediction result and B represents the true label. It also does not require complex logistic calculation and reflects a more intuitive spatial overlap, which is one of the core metrics for evaluating the capability of segmentation models. In the research, it is used to validate the model training combined with Dice coefficient.

2.3. Implementation details

The experience was implemented on the Kaggle platform. The system was running Python version 3.6.6, with the PyTorch version 1.2.0 installed. It was operating on a Linux 6.6.56+ kernel. The Central Processing Unit (CPU) is an Intel Xeon at 2.00 GHz, and the GPU is a Tesla P100-PCIE-16GB. The optimizer used is Adam with a learning rate of 1e-3, and the learning rate scheduler is Reduce Learning Rate on Plateau, configured to operate in "min" mode with a patience of 3. The system is also using the Albumentations library of version 1.0.3, and the batch size is set as 4. Each group was trained for 30 epochs, after which, the metrics of the groups reached a stable level.

3. Results and discussion

In this section, the final metrics after the application of different kinds of data augmentation methods are analyzed. The analysis is done from the perspective of overfitting by observing the curves during the training process. The data obtained is synthesized for further discussion.

3.1. Eventual training metrics

Surprisingly, the validation result showed that the groups using data augmentation did not lead to a significant improvement in the training as is usually the case compared to the group that did not utilize the methods. Instead, some of the data augmentation caused a decrease in the results. As Table 2 showed, the group that did not apply data augmentation methods achieved one of the best validation results, with the validation Dice coefficient of 0.7626 and IoU of 0.5215. In contrast, the methods related to flipping and color transformations did not cause a significant improvement in the validation of the training, whose metrics of Dice coefficient in validation and IoU in validation are similar to the baseline value in the eventual result. Other data augmentation methods made the final training results worse in varying degrees instead of improving the results.

Table 2: Metrics of each group

Group Number	Method	Training: Dice coefficient	Training: IoU	Validation: Dice coefficient	Validation: IoU
1	Baseline	0.8997	0.7435	0.7626	0.5215
2	Flip	0.8101	0.5895	0.7674	0.5183
3	Geometric Transformation	0.6668	0.4021	0.6219	0.4636
4	Degradation	0.6410	0.3412	0.7137	0.4593
5	Structural Disruption	0.6479	0.3788	0.7126	0.4545
6	Color Space Change	0.8498	0.6740	0.7566	0.5204
7	Nature Simulation	0.5772	0.2737	0.6721	0.3864
8	Size Adjustment	0.8522	0.3858	0.6434	0.3738

After selecting the two best-performing data augmentation methods, the experiments were conducted again after adjusting their parameters or recombining them. For Flip methods, the probabilities of Horizontal Flip/Vertical Flip are modified to 0.75/0.25, 0.25/0.75 and 0.25/0.25 respectively. The results obtained are shown in Table 3. After adjustments or recombination, there are not obvious improvements compared to the baseline.

Table 3: Metrics of adjusted group

Group Number	Method	Training: Dice coefficient	Training: IoU	Validation: Dice coefficient	Validation: IoU
1	Flip (0.75/0.25)	0.8003	0.5733	0.7562	0.4969
2	Flip (0.25/0.75)	0.7665	0.5156	0.7266	0.4583
3	Flip (0.25/0.25)	0.7934	0.5735	0.7607	0.4977
4	Flip (0.5 /0.5), Color Space Change	0.7835	0.5488	0.7462	0.4842

This is possibly related to the effect of different augment methods on the original image. Taking 5b202c6f5.jpg as an example, as Figure 4 showed, the Flip only affects the location of the defect and has no effect on other factors. Geometric Transformation and Size Adjustment causes some of the labeled locations to be cropped, which reduces the critical information in the image. Degradation and Nature Simulation leads to an increase in the amount of noise on the picture, which can make the border between the defect and the background less visible than the baseline. Structural Disruption causes the picture to be distorted, which produces a different result than a normal defect. Color Space Change affects the contrast of the picture, but the most important information is still preserved. Therefore, it's normal that Flip and Color Space Change did not cause performance degradation even though they did not improve the final result. This phenomenon might be attributed to the fact that the dataset's size is already sufficient to meet the requirements of model training; thereby, the choice of

method does not have a significant impact on the final results. However, other methods mentioned above will cause the key information of the image to be obscured, interfered with, or distorted, which will make the final result worse.

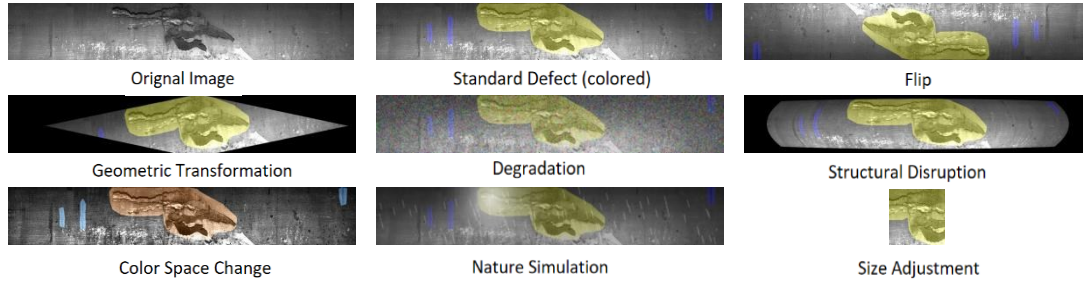


Figure 4: Impact on image of the augment methods (picture credit: original)

3.2. Overfitting situation

Comparing several of the best training curves, as in Figure 5, it is obvious that even though data augmentation methods did not result in a significant optimization in terms of the final metric values, they resulted in a higher degree of correspondence between the training and validation curves. And it is clear that the Dice coefficients and IoUs in the training curves are getting significantly higher than those in the validation curves without data augmentation since the 14th epoch, while the group with Combination and Flip groups did not show significant overfitting phenomena. The comparison also proves that even though data augmentation cannot improve the final result in some circumstances, it can still prevent overfitting to a certain extent.

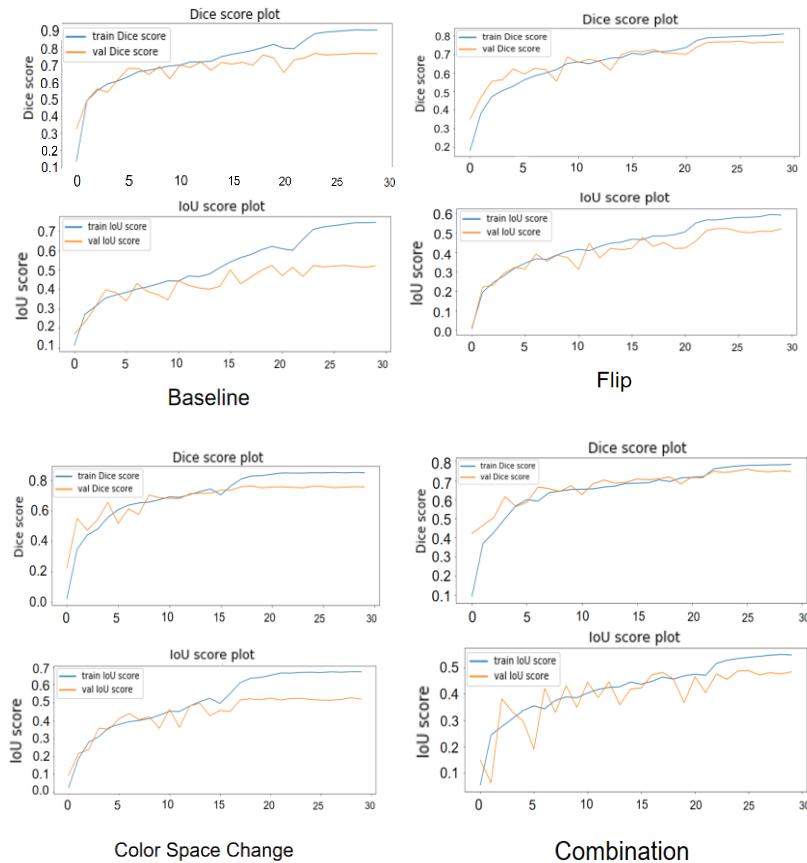


Figure 5: Metric curves with different augment methods (picture credit: original)

From the data of result, most of the data augmentation methods do not cause significant results, while some of them may even lead to a decrease in the results. Therefore, data augmentation methods should be chosen carefully for data sets with important information. Although some methods cannot make the final results stronger, their contribution in preventing overfitting cannot be denied. The number of epochs can be reduced by choosing appropriate methods, which can be used for optimization of time.

4. Conclusion

This research focuses on the impact of data augmentation on model training for segmentation tasks of steel surface defect detection. During the study, U-Net was used as the main architecture, and ResNet-18 was used as its encoder. The study explored the effects of data augmentation methods by replacing different operations and adjusting the parameters. In this result, the application of data augmentation methods did not cause significant improvement in the validation metrics. On the contrary, some methods made the results worse, especially the methods that can make important information obscured or interfered with or make the defects severely deformed. However, by reasonably utilizing data augmentation methods, it is undeniable that the occurrence of overfitting can still be reduced to a certain extent. However, the study also has certain limitations. Only one single dataset was used in the experiment, and due to the wide variety of data augmentation methods, they were categorized only according to the operational purpose of the methods. Moreover, the hyperparameters of each method are seldom modified, which probably makes sense for the result. In the future, research will employ more diverse datasets as well as refined adaptation of different method groups for further exploration.

References

- [1] Motallebi, S., Zandieh, M., Tabriz, A.A., & Tirkolaee, E.B. (2025) *Assessing the industry 4.0 strategies for a steel supply chain: SWOT, game theory, and gap analysis*. *Heliyon*, 11(1).
- [2] Damacharla, P., Rao, A., Ringenberg, J., & Javaid, A. Y. (2021) *TLU-net: a deep learning approach for automatic steel surface defect detection*. *International Conference on Applied Artificial Intelligence*, 1-6.
- [3] Devlin, A., Mykhnenko, V., Zagoruichyk, A., Salmon, N., & Soldak, M. (2024) *Techno-economic optimisation of steel supply chains in the clean energy transition: A case study of post-war Ukraine*. *Journal of Cleaner Production*, 466, 142675.
- [4] Tang, B., Chen, L., Sun, W., & Lin, Z.K. (2023) *Review of surface defect detection of steel products based on machine vision*. *IET Image Processing*, 17(2), 303-322.
- [5] Liu, X., & Zhou, Y. (2024). *Surface Defect Detection on Steel Plates Based on Improved Unet Network*. *Chinese Control and Decision Conference*, 6268-6273.
- [6] Bai, D., Li, G., Jiang, D., Tao, B., Yun, J., Hao, Z., ... & Ju, Z. (2024) *Depth feature fusion based surface defect region identification method for steel plate manufacturing*. *Computers and Electrical Engineering*, 116, 109166.
- [7] Wang, Z., Wang, P., Liu, K., Wang, P., Fu, Y., Lu, C.T., ... & Zhou, Y. (2024) *A comprehensive survey on data augmentation*. *arxiv preprint*, 2405.09591.
- [8] Kaggle dataset. (2022). *Severstal: Steel Defect Detection*. Retrieved from <https://www.kaggle.com/competitions/severstal-steel-defect-detection/data>.
- [9] Krithika, A.A.M., & Suganthi, K. (2022) *Review of semantic segmentation of medical images using modified architectures of UNET*. *Diagnostics*, 12(12), 3064.
- [10] Siddique, N., Paheding, S., Elkin, C.P., & Devabhaktuni, V. (2021) *U-net and its variants for medical image segmentation: A review of theory and applications*. *IEEE access*, 9, 82031-82057.
- [11] He, K., Zhang, X., Ren, S., & Sun, J. (2016) *Deep residual learning for image recognition*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.
- [12] Kang, Z., Xiao, E., Li, Z., & Wang, L. (2024) *Deep learning based on ResNet-18 for classification of prostate imaging-reporting and data system category 3 lesions*. *Academic Radiology*, 31(6), 2412-2423.
- [13] Sun, K.S., Sangam, A., Suchetha, M., Raman, R., Rajalakshmi, R., & Tamilselvi, S. (2024) *A refined ResNet18 architecture with Swish activation function for Diabetic Retinopathy classification*. *Biomedical Signal Processing and Control*, 88, 105630.