Rethinking Regularization with Random Label Smoothing

Claudio Filipi Goncalves dos Santos (cfsantos@ufscar.br)
Federal University of São Carlos

Joao Paulo Papa
São Paulo State University

Research Article

Keywords: Convolutional Neural Networks, Regularization, Label Smoothing

Posted Date: July 1st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1780619/v1

Rethinking Regularization with Random Label Smoothing

Claudio Filipi Gonçalves dos Santos[1,2*] and João Paulo Papa[3]

[1*]Department of Computer Science, Federal University of Sao Carlos - UFSCar, Washington Luiz Road, Sao Carlos, 13565-905, SP, Brazil.
[2]Department of Software Application, Eldorado Research Institure, 275 Alan Turing Av., Campinas, 13083-898, SP, Brazil.
[3]Department of Computing, State University of Sao Paulo - UNESP, 14-01 Eng. Luís Edmundo Carrijo Coube Av., Bauru, 17033-360, SP, Brazil.

*Corresponding author(s). E-mail(s): cfsantos@ufscar.br,claudio.santos@eldorado.org.br;
Contributing authors: joao.papa@unesp.br;

Abstract

Regularization helps to improve machine learning techniques by penalizing the models during training. Such approaches act in either the input, internal, or output layers. Regarding the latter, label smoothing is widely used to introduce noise in the label vector, making learning more challenging. This work proposes a new label regularization method, Random Label Smoothing, that attributes random values to the labels while preserving their semantics during training. The idea is to change the entire label into fixed arbitrary values. Results show improvements in image classification and super-resolution tasks, outperforming state-of-the-art techniques for such purposes.

Keywords: Convolutional Neural Networks RegularizationLabel Smoothing

1 Introduction

Neural networks are acknowledged to be learn-by-example techniques. Assuming the training set is representative, the problem becomes finding proper loss functions to avoid local optima and a good backbone. We know that some Convolutional Neural Networks (CNNs) are more accurate than others. In image classification tasks, ResNet He et al (2016a) usually performs better than VGG Simonyan and Zisserman (2014), for it has a more complex architecture and residual connections that help generalization He et al (2016b).

Changing the training protocol is another approach to increase generalization. Notably, neural networks can obtain better results when more (and proper) training instances are available. One can also train a model in some larger dataset before fine-tuning it to the desired problem, i.e., transfer learning Ribani and Marengoni (2019). Results show it can boost the outcomes significantly in a variety of problems. Google improved results in the ImageNet challenge Deng et al (2009) by creating a huge dataset with more than 300 million images to first train a model and further fine-tune it in the ImageNet set Sun et al (2017).

However, some scenarios do not allow us to collect additional training data, for the labeling cost is prohibitive. Regularization can come to this aid by either making training harder or the loss function landscape smoother Goodfellow et al (2016). We expect to achieve improvements in the model's generalization after the application of such approaches. Classical regularization approaches in deep learning include data augmentation, which shall consider semantics after transformation. A model that trains on the MNIST dataset LeCun et al (2010), for instance, can rotate images to some extent only. Rotating a "6" in 180 degrees ends up in a "9", generating a wrong instance-label pair.

This work introduces Random Label Smoothing (RLS), a new regularization approach that works on the output layer. RLS operates by randomly changing values in the label vector (ground truth). We demonstrate state-of-the-art results in image classification and super-resolution, evidencing it can be used in a broad range of application domains.

The manuscript is organized as follows. Sections 2 and 3 present some related works and the proposed approach, respectively. Section 4 introduces the methodology, and Section 5 demonstrates the robustness of the proposed approach in experiments under different scenarios. Section 6 provides a brief discussion about the outcomes, and Section 7 states conclusions.

2 Related Works

Several regularization techniques are available for helping neural networks to accomplish better results in different domains. Regularization based on data augmentation is classic and with many approaches in the literature. AutoAugment Cubuk et al (2018) performs data augmentation by first learning the best policy for creating synthetic samples. However, it may take too long to determine the best data augmentation strategy for a given data set. Aiming

to make this process faster, Fast AutoAugment Lim et al (2019) calculates the gradient from just one batch, decreasing the computational effort considerably. Other methods, such as Cutout DeVries and Taylor (2017) and RandomErasing Zhong et al (2020), work by removing random areas of the image. The former removes a patch and leaves its content empty, while the other fills it with some random noise.

Other methods operate by changing the feature maps generated during training. Dropout Srivastava et al (2014) randomly drops neurons, while MaxDropout Santos et al (2020) eliminates the most active ones, i.e., the neurons with the highest activation values. An improved version, called MaxDropoutV2 Santos et al (2022), includes a more efficient approach to finding the most active neurons. Instead of directly comparing values on the output feature map from a given layer, it first sums the value of each neuron in the depth axis for further performing the comparison. MaxDropoutV2 carries more semantic information than its original counterpart. Additional methods consider other internal aspects when training CNNs. Shake-Shake Gastaldi (2017) changes the weights of the inference and the backpropagation values on training time in a multi-branch model, such as ResNeXT Xie et al (2017). Results show it can significantly improve the results.

A recent analysis of regularization methods for CNNs Santos and Papa (2022) raised some interesting drawbacks in the area. The first one is the shortage of algorithms that perform regularization on a label level. The other point concerns the application domain, i.e., most regularization techniques designed for deep nets focus on image classification. Label Smoothing Szegedy et al (2016) changes the values of the output layer (label vector), i.e., it decreases the value of the position that represents the true label and increases the values of the inactive labels. The Two-Stage Label Smoothing (TSLA) Xu et al (2020) changes the label values to some extent during training. The work shows that stopping label smoothing in a late training stage helps the model to generalize better.

3 Proposed Approach

According to Santos and Papa (2022), there are several issues related to some new regularization methods. We address quite a few of them in this work. Regularization approaches are often evaluated in a single context only, primarily on image classification. Here, we also consider image super-resolution. Still, according to to Santos and Papa (2022), a good regularization technique should improve results even if the model is already using another regularization technique, which RLS is capable of.

Traditional data augmentation usually changes features in the input data. A simple way to perform data augmentation is to rotate the image to the left or to the right in random degrees. Another way is to crop some areas of the input image, e.g., Cutout DeVries and Taylor (2017). Following a similar logic, RLS augments the data by performing random but controlled changes in the

	Batch - Regular Training			Batch - Label Smoothing			Batch - Random Label Smoothing		
Class 0	0	0	0	0.022	0.022	0.022	0.1	0.05	0.02
Class 1	0	0	0	0.022	0.022	0.022	0.05	0	0.15
Class 2	0	0	0	0.022	0.022	0.022	0.03	0.03	0.05
Class 3	0	**1**	0	0.022	**0.8**	0.022	0.17	**0.75**	0.01
Class 4	0	0	0	0.022	0.022	0.022	0.02	0.02	0
Class 5	**1**	0	0	**0.8**	0.022	0.022	**0.5**	0.06	0.03
Class 6	0	0	0	0.022	0.022	0.022	0.03	0.01	0.01
Class 7	0	0	0	0.022	0.022	0.022	0.04	0.03	0.03
Class 8	0	0	**1**	0.022	0.022	**0.8**	0.01	0.04	**0.68**
Class 9	0	0	0	0.022	0.022	0.022	0.05	0.01	0.02

Fig. 1 Simulation of Label Smoothing and Random Label Smoothing over a batch of labels during training for a classification model (active label in bold). Traditionally, the active label is set to"1" while all other classes' indices are set to '0". In Label Smoothing, the active class is set to a constant (and higher) value, while the inactive classes are set to a smaller invariant value. In Random Label Smoothing, the active label receives a random (and higher) value (e.g., greater than 0.5) while the inactive labels receive a random value that, summed with the active label value, reaches 1.

label of every single instance during training. We explain how to do that for image classification and super-resolution tasks.

3.1 Image Classification

Concerning image classification, we vary the output values that define the label in a controlled but random range of values. In the "active" position, i.e., the index represented by the value "1" that encodes the label (one-hot representation), we randomly decrease its values between 0.05 and 0.49, guaranteeing that the active label will always have the greatest value. For the inactive positions (defined as "0"), we divided the amount of value used earlier among these positions. For instance, if the problem has 10 classes and the removed value is 0.3, the active position is set to 0.7, and all the other 9 positions receive a portion of the remaining value.

By doing these transformations in the labels during training, we create various acceptable labels for a given instance, working as an augmented label algorithm. Our results show it is functional for image classification, helping the model to generalize better and overcoming other methods, such as TargetDrop Zhu and Zhao (2020) and MaxDropout Santos et al (2020). Figure 1 demonstrates how RLS works for a classification problem in a toy example.

3.2 Image Super-Resolution

For image reconstruction, we first tried a similar approach to the classification task by randomly changing the label's values following a Gaussian distribution[1]. However, it did not work as expected. The new reference image (modified ground truth) figures much Gaussian noise by changing the pixel values using a Gaussian distribution. The entire system then learns to reconstruct images with noise.

[1]Now, the pixel values encode the labels.

Changing the values pixel-by-pixel with different random values did not seem to be a good strategy, for we lose semantic details. We, therefore, decided to perturb all pixels by the same amount, i.e., a random value that can be either added or subtracted by the pixel value in a given training interaction. The problem is now defining what values range leads to the best results.

We achieved promising outcomes by reverse-engineering the results of the neural networks employed in this study. We used the results (i.e., PSNR - peak signal-to-noise ratio) of each architecture to set a range of values with better results. For instance, PyNET Ignatov et al (2020) achieved a 21.19 dB of PSNR; then, converting this value to a gray-scale amount results in about 12 units. Therefore, all pixel values (for all dataset images) were either subtracted or added concerning that amount (in this example). For EDSR (Enhanced Deep Residual Networks for Single Image Super-esolution) Lim et al (2017), the results are a PSNR of 29.21 dB for Div2k Agustsson and Timofte (2017) and 28.89 dB for the RealSR dataset Cai et al (2019). Therefore, we set 4.5 units for both datasets.

We verified if the above methodology could be further improved in the last experiment. We found out that we could achieve even better results by using half of the range than using the full range. In this case, we used 6 units for the PyNet and 2.25 for EDSR. Even though the image regions may be different, smaller differences in continuous regions can help the entire model understand that smaller errors are more acceptable than the exact value.

4 Methodology

This section provides a complete description of the experimentation protocol we used to evaluate RLS. We divided the experiments into three main parts: first, all four architectures we used for evaluating purposes are described. Right after, we presented the training protocol and later a description of the datasets.

4.1 Scenarios

We considered three different scenarios to evaluate RLS. They all have, in some ways, differences in the input data or the label level. The first one is standard image classification, and the second task concerns image super-resolution. In this case, the neural network's task is to magnify the image input, creating another but amplified. For example, for a magnification of four times, if the input has the size of 200×200, the model's objective is to create an output of size 800×800.

Last but not least, another challenge is to simulate an image signal processor (ISP), which basically creates an RGB image from a CFA (color filter array) acquired by the camera's sensor. Therefore, given a CFA input, the task is to learn a CNN that can generate its corresponding RBG output.

4.2 Neural Network Architecture

As mentioned earlier Santos and Papa (2022), a good regularization technique should improve results in different problems to show it can enhance a given CNN outcome. We tested four neural backbones in different neural architectures to provide a fair evaluation: two for image classification, one for single image super-resolution, and one for software ISP.

The first CNN we use to evaluate RLS is ResNet He et al (2016a), more precisely, ResNet-18. We have chosen this architecture because it is widely used for evaluating regularization techniques, allowing a natural comparison. Such neural backbone comprises a sequence of convolutional and pooling layers, with pooling after a sequence of two or three convolutional layers. The significant innovation in its architecture concerns the residual connections, which may improve effectiveness to a certain extent.

EDSR Lim et al (2017) is one of the scarce neural networks used to evaluate regularization methods Yoo et al (2020), ending up in another natural choice. It stands for a residual convolutional network with a sequence of convolutional-ReLU activation-convolutional operations in its residual blocks and pixel shuffle operations Shi et al (2016) to perform image super-resolution in the end. PyNET Ignatov et al (2020), a multi-branch CNN that has several layers in parallel and uses different measures for error calculation, is interesting in evaluating problems related to image and signal processing, specifically image reconstruction.

4.3 Training Protocol

For the image classification problem, we considered the protocol suggested by Santos and Papa (2022). The images were redimensioned to 32×32 pixels and then randomly cropped in 28×28 patches. Stochastic gradient descent with Nesterov momentum is used for gradient calculation. The learning rate starts at $10e-2$ and is multiplied by $10e-1$ on epochs 80, 120, and 160.

Concerning image super-resolution, we did not find any defined or suggested protocol. We, therefore, followed the same parameters used in CutBlur Yoo et al (2020) and PyNET Ignatov et al (2020). We understand a natural comparison to these previous works by observing the same parameters.

In all scenarios, five training runs were performed to avoid comparing results only by chance. Our results report the mean and standard deviation values for each performance measure.

4.4 Datasets

We used a different dataset for each task evaluated in this work to allow a fair comparison against other methods. In each case, we selected datasets that, according to our research, are the most used ones on each application domain considered here, i.e., image classification and super-resolution.

For the image classification task, we appointed CIFAR-100 Krizhevsky et al (2009), one of the most used datasets to evaluate regularization techniques Santos and Papa (2022). It comprises 50,000 images from 100 different classes for training purposes and 10,000 images as the validation set.

For the single image super-resolution task, we considered two different datasets inspired in Yoo et al (2020). The first one stands for the Div2K Agustsson and Timofte (2017) dataset, with 800 pairs of low and high-resolution RGB images for training and 100 for evaluation. The other dataset is RealSR Cai et al (2019), which comprises 459 pairs of images for training and 100 for model validation. We used a magnification of four times for comparison purposes in both cases.

The last one is the Zurich RAW to RGB Dataset Ignatov et al (2020), which evaluates techniques for image reconstruction. This dataset is divided into 46,839 pairs of RGB Bayer filter data/RGB image for training and 1,204 similar pairs for testing purposes.

5 Experimental Results

This section provides outcomes of RLS against some state-of-the-art regularization approaches. RLS is first evaluated over image classification tasks (Section 5.1) and later on image super-resolution problems (Section 5.2).

5.1 Image Classification

Table 1 presents the error rate concerning ResNet-18 in CIFAR-100 dataset. RLS has achieved the best outcome solely, outperforming seven other techniques. The first row in the table is our baseline, i.e., ResNet-18, without any regularization.

Method	Error (%)
ResNet-18 Zhong et al (2020)	24.50
Cutout DeVries and Taylor (2017)	21.96
RandomErasing Zhong et al (2020)	24.03
MaxDropout Santos et al (2020)	21.93
MaxDropoutV2 Santos et al (2022)	21.92
TSLA Xu et al (2020)	21.45
TargetDrop Zhu and Zhao (2020)	21.45
RLS	$\mathbf{21.18 \pm 0.35}$

Table 1 Image classification experiment over CIFAR-100 dataset.

5.1.1 Working Along with Other Regularizers

As mentioned by Santos and Papa (2022), it is vital to check how a particular regularization algorithm works along with other regularization methods. Here, we provide some interesting outcomes. Table 2 presents results of ResNet-18 using Cutout and other methods working together. The proposed approach

can outperform MaxDropout and TargetDrop working jointly with Cutout by more than 0.5% on average, which is to be considered a good improvement.

Method	Error (%)
ResNet-18 Zhong et al (2020)	24.50
Cutout DeVries and Taylor (2017)	21.96
MaxDropout + Cutout Santos et al (2020)	21.82
MaxDropoutV2 + Cutout Santos et al (2020)	21.82
TargetDrop + Cutout Zhu and Zhao (2020)	21.25
RLS + Cutout	**20.6 ± 0.16**

Table 2 Results on CIFAR-100 using ResNet-18 with one or more regulization methods.

Combining PyramidNet with ShakeDrop and RLS results in some improvement too. Table 3 shows the outcomes of PyramidNet without any regularization, using ShakeDrop, and using ShakeDrop+RLS. On average, that combination allowed an improvement of 0.1%.

Method	Error (%)
PyramidNet Han et al (2017)	16.35
ShakeDrop Yamada et al (2019)	16.22
ShakeDrop + RLS	**16.12 ± 0.14**

Table 3 Results on CIFAR-100 using PyramidNet with one or more regulization methods.

5.2 Image Super-Resolution

Table 4 shows the outcomes of EDSR backbone using different regularization techniques from Yoo et al (2020). Some conclusions can be drawn in this scenario. The first one concerns Div2k dataset, whose results show that RLS using half of the perturbation value (RLS-Half) outperforms all methods, including the situation when all techniques (i.e., EDSR, Cutout, Cutmix, Mixup, RGB permutation, Blend, and Cutblur) are used together (All).

The second analysis concerns the outcomes of the RealSR dataset. Although RLS did not overcome the neural network trained using all methods, still, it has the best individual result.

We considered an additional experiment related to image reconstruction. Table 5 shows the outcomes of PyNET using RLS algorithm considering PSNR and Multiscale Structural Similarity Index Measure (MS-SSIM) Wang et al (2003) quality measures. An improvement in the original results (i.e., standard PyNet) can be observed when RLS is applied. It is worth mentioning that, even though there are several regularization methods available for CNNs, none of them tackles, or at least is evaluated, in the context of image reconstruction. As far as we are aware Santos and Papa (2022), this is the first regularization approach that improves the results of deep learning models in the aforementioned task.

	Div2K	RealSR
EDSR	29.21	28.89
Cutout	29.22 ± 0.01	28.95 ± 0.06
Cutmix	29.22 ± 0.01	28.89 ± 0.00
Mixup	29.26 ± 0.05	28.98 ± 0.09
CutMixup	29.27 ± 0.06	29.03 ± 0.14
RGB Permutation	29.30 ± 0.09	29.02 ± 0.13
Blend	29.23 ± 0.02	29.03 ± 0.14
Cutblur	29.26 ± 0.05	29.12 ± 0.23
All	29.30 ± 0.30	$\mathbf{29.16 \pm 0.27}$
RLS-Gaussian	28.03 ± 0.07	27.97 ± 0.04
RLS-Full	29.31 ± 0.01	29.05 ± 0.04
RLS-Half	$\mathbf{29.32 \pm 0.01}$	29.15 ± 0.03

Table 4 PSNR results on Div2K and RealSR datasets for EDSR using regulization methods.

Method	PSNR	MS-SSIM
PyNet Ignatov et al (2020)	21.19	0.862
Pynet + Gaussian-RLS	20.86	0.850
Pynet + Full-RLS	21.21	0.863
Pynet + Half-RLS	$\mathbf{21.22 \pm 0.01}$	**0.867**

Table 5 Results on Zurich RAW to RGB Dataset for PyNet.

6 Discussion

Providing new regularization algorithms is not straightforward for it often needs the knowledge of a specialist in the problem. Deep learning by itself is already an area of research that demands plenty of work when some improvements are required. This section provides some discussion about the outcomes obtained in the previous section.

6.1 Lack of Label Regularization Methods

Achieving the best results using a neural network is always desired, and regularization methods should be encouraged in most cases, as far as it does not break the semantics of the dataset. The label can be considered safe to test for multi-class classification, regardless of the application domain. The output is often hot-encoded, so there are few possibilities to harm or lose semantics.

This work is about a regularization method for Convolutional Neural Networks. It is fair and easy to compare with other algorithms because it follows the same evaluation protocol. However, more techniques rather than TSLA are desired to perform a better and direct comparison.

6.2 Lack of Comparison for General-purpose Applications

There might be a bias toward creating deep learning regularization algorithms only for the image classification task. We could find several regularization methods DeVries and Taylor (2017); Santos et al (2020); Zhu and Zhao (2020); Zhong et al (2020) for comparison purposes; however, we found only one

for directly comparing in the context of image super-resolution Yoo et al (2020). Besides, as far as we are aware, no other in-depth study compared regularization techniques for image reconstruction.

The scarcity of works that aimed to compare regularization techniques in problems other than image classification is worrying. Indeed, we found some works Yoo et al (2020); Santos and Papa (2022) that also complain about this scarcity of research on regularization algorithms in more problems. We encourage the researchers to develop new methods for other image processing problems, for there might be a promising area of research.

7 Conclusions and Future Works

We presented the RLS technique for label-level regularization concerning Convolutional Neural networks. Our results demonstrate that it can outperform other techniques when applied to different image processing problems. As such, we tackle not only the enhancement of neural networks but the problem of generalizing regularization algorithms, complained by Santos and Papa (2022). RLS can be combined with other techniques and be used within any backbone.

We intend to apply RLS to other problems in future works, such as natural processing language processing. Another intent is to check if there are random distributions that can improve our current results.

Acknowledgments

The authors are grateful to the São Paulo Research Foundation (grants 2013/07375-0, 2014/12236-1, and 2019/07665-4), to Eldorado Research Institute, and to the National Council for Scientific and Technological Development (grant 308529/2021-9).

References

Agustsson E, Timofte R (2017) Ntire 2017 challenge on single image super-resolution: Dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops

Cai J, Gu S, Timofte R, et al (2019) Ntire 2019 challenge on real image super-resolution: Methods and results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops

Cubuk ED, Zoph B, Mane D, et al (2018) Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:180509501

Deng J, Dong W, Socher R, et al (2009) ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09

DeVries T, Taylor GW (2017) Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:170804552

Gastaldi X (2017) Shake-shake regularization. arXiv preprint arXiv:170507485

Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press, http://www.deeplearningbook.org

Han D, Kim J, Kim J (2017) Deep pyramidal residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5927–5935

He K, Zhang X, Ren S, et al (2016a) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

He K, Zhang X, Ren S, et al (2016b) Identity mappings in deep residual networks. In: European conference on computer vision, Springer, pp 630–645

Ignatov A, Van Gool L, Timofte R (2020) Replacing mobile camera isp with a single deep learning model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp 536–537

Krizhevsky A, Nair V, Hinton G (2009) Cifar-10 and cifar-100 datasets. URl: https://www cs toronto edu/kriz/cifar html 6:1

LeCun Y, Cortes C, Burges C (2010) Mnist handwritten digit database. att labs

Lim B, Son S, Kim H, et al (2017) Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 136–144

Lim S, Kim I, Kim T, et al (2019) Fast autoaugment. In: Advances in Neural Information Processing Systems, pp 6665–6675

Ribani R, Marengoni M (2019) A survey of transfer learning for convolutional neural networks. In: 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), IEEE, pp 47–57

Santos CFG, Papa JP (2022) Avoiding overfitting: A survey on regularization methods for convolutional neural networks. ACM Comput Surv https://doi.org/10.1145/3510413, URL https://doi.org/10.1145/3510413, just Accepted

Santos CFGd, Colombo D, Roder M, et al (2020) Maxdropout: Deep neural network regularization based on maximum output values. In: Proceedings of 25th International Conference on Pattern Recognition, ICPR 2020, Milan, Italy, 10-15 January, 2021. IEEE Computer Society, pp 2671–2676

Santos CFGd, Roder M, Passos LA, et al (2022) Maxdropoutv2: An improved method to drop out neurons in convolutional neural networks. In: Iberian Conference on Pattern Recognition and Image Analysis, Springer, pp 271–282

Shi W, Caballero J, Huszár F, et al (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1874–1883

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

Srivastava N, Hinton G, Krizhevsky A, et al (2014) Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15(1):1929–1958

Sun C, Shrivastava A, Singh S, et al (2017) Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision, pp 843–852

Szegedy C, Vanhoucke V, Ioffe S, et al (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826

Wang Z, Simoncelli EP, Bovik AC (2003) Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Ieee, pp 1398–1402

Xie S, Girshick R, Dollár P, et al (2017) Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1492–1500

Xu Y, Xu Y, Qian Q, et al (2020) Towards understanding label smoothing

Yamada Y, Iwamura M, Akiba T, et al (2019) Shakedrop regularization for deep residual learning. IEEE Access 7:186,126–186,136

Yoo J, Ahn N, Sohn KA (2020) Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 8375–8384

Zhong Z, Zheng L, Kang G, et al (2020) Random erasing data augmentation. In: AAAI, pp 13,001–13,008

Zhu H, Zhao X (2020) TargetDrop: A targeted regularization method for convolutional neural networks