PLOS ONE

RESEARCH ARTICLE

# MeshCut data augmentation for deep learning in computer vision

**Wei Jiang**[1], **Kai Zhang**[1], **Nan Wang**[2]*, **Miao Yu**[1]*

**1** School of Mechanical Engineering, Sichuan University, Chengdu, China, **2** School of Physical Science and Technology, Southwest Jiaotong University, Chengdu, China

* 705679317@qq.com (NW); miaoyu@scu.edu.cn (MY)





**Citation:** Jiang W, Zhang K, Wang N, Yu M (2020) MeshCut data augmentation for deep learning in computer vision. PLoS ONE 15(12): e0243613. https://doi.org/10.1371/journal.pone.0243613

**Editor:** Zhishun Wang, Columbia University, UNITED STATES

**Received:** July 12, 2020

**Accepted:** November 24, 2020

**Published:** December 23, 2020



**Data Availability Statement:** All relevant data are within the paper and its Supporting Information files.

**Funding:** The project of Sichuan Province Science and Technology Support Program under Grant 2019YFG0373; The Major Special Projects of Sichuan Province under Grant 2020ZDZX0024; Department of Science and Technology of Sichuan Province, 2019YFG0397.

**Competing interests:** The authors have declared that no competing interests exist.

## Abstract

To solve overfitting in machine learning, we propose a novel data augmentation method called MeshCut, which uses a mesh-like mask to segment the whole image to achieve more partial diversified information. In our experiments, this strategy outperformed the existing augmentation strategies and achieved state-of-the-art results in a variety of computer vision tasks. MeshCut is also an easy-to-implement strategy that can efficiently improve the performance of the existing convolutional neural network models by a good margin without careful hand-tuning. The performance of such a strategy can be further improved by incorporating it into other augmentation strategies, which can make MeshCut a promising baseline strategy for future data augmentation algorithms.

## Introduction

Recently, convolutional neural networks (CNNs) have demonstrated massive potential in the field of computer vision [1–12]. Modern CNNs commonly contain millions of parameters to try to achieve sufficient representational power for difficult tasks, but the excessive representational power inevitably risks overfitting, resulting in poor generalisation.

Data augmentation is an effective regularization strategy [13] by which more training samples can be generated to alleviate overfitting. Thereinto, geometric transforms and photometric distortions are two widely used methods. For geometric transforms, the basic operations include random scaling, flipping, rotating, and cropping. For photometric distortions, the operations include contrast, brightness, and hue. Although the two methods are pixel-level adjustments, the original information is left intact. Another type of data augmentation method simulates object occlusion, which can help the network learn relatively weak target features and improve the perception ability, thus increasing the network's robustness. Typically, random erasing [14] and CutOut [15] randomly zero out one continuous block region in the input image. The HaS [16] and GridMask [13] methods randomly or evenly remove several block regions in an image. Expanding those concepts by applying them to feature maps are DropConnect [17], DropOut [18], and DropBlock [19]. Other proposed methods use data augmentation based on multisource image fusion; for example, MixUp [20] and CutMix [21]. Of

these, MixUp superimposes two images with different pellucidity, and CutMix fills a cropped image with the rectangle regions of other images.

Among the abovementioned methods, the information occlusion methods, as one of the most promising methods, have demonstrated their specific ability to prevent the overfitting. However, there exist some deficiencies in practical use, such as the following: 1) The overfocused partial information dropping could not be inherently avoided, resulting in the incompatibility for small objectives. 2) Little attention was paid to the occlusion of context information, leading to the weak ability to learn the partial features. 3) The information confusion could not be effectively simulated, which was disadvantageous for further improving the robustness.

In the recent studies, some CNNs [22, 23], in combination with the recurrent neural network (RNN), have been used in an attempt to capture more target features that are inherent in a video sequence. These excellent methods provide a multi-state representation for the target and module the uncertain transformation for the background; hence, their performance in terms of accuracy is improved tremendously. Inspired by these ideas, we are intrigued to observe that global information fragmentisation has some similar effects, which is helpful for solving the three abovementioned problems. On this basis, we propose a novel augmentation method named MeshCut.

## MeshCut

MeshCut is a simple, universal, and effective strategy that can be easily implemented in most of the existing CNN models. Different from previous methods, MeshCut, by superimposing a mesh mask in the training phase, transforms an image into a mosaic made with several image fragments, as shown Fig 1. Because of the zeroed-out margins between fragments, the overall feature of the target is effectively broken up into several partial features, which can provide more diversified information for the network.

The MeshCut processing flow is shown in Fig 2. During the training phase, MeshCut overlays a binary mask M on the input image, namely

$$\mathrm{I}_{\mathrm{out}} = I_{in} \times M, \tag{1}$$

where $I_{in} \in R^{H\times W\times C}$ is the input image, $M \in R(0, 1)^{H\times W}$ is the mask, and $I_{out} \in R^{H\times W\times C}$ is the output image with information occlusion.

Fig 3 shows the mesh mask. For pixel(i,j) with a grey value of 1 in mask M, the corresponding regions of input I remain, but for pixel(i,j) with a grey value of 0, the corresponding regions are zeroed out. This operation should be applied after the image normalization operation.

Three parameters (p, wr and $m_{prob}$) are introduced for controlling the shape of mask M, where p is the line-spacing, wr is the line width, and $m_{prob}$ is the intervention rate of the overlaying operation. For p, some randomness is usually added to extend the applicability to a variety of images, which can be written as follows

$$\mathrm{p} = random(p_{max}, p_{min}), \tag{2}$$

For wr, it is a normalized ratio to the line spacing p in practice, which could be written as

$$\mathrm{wr} = pixels_{wr}/pixels_{p}, \tag{3}$$

where $pixels_{wr}$ and $pixels_{p}$ are the number of pixels corresponding to the line spacing and the line width respectively.

To increase the randomness of segmentation, the entire mask is shifted by a random distance within a range of ±p/2 and rotated by a random angle within a range of 0˚ to 360˚.

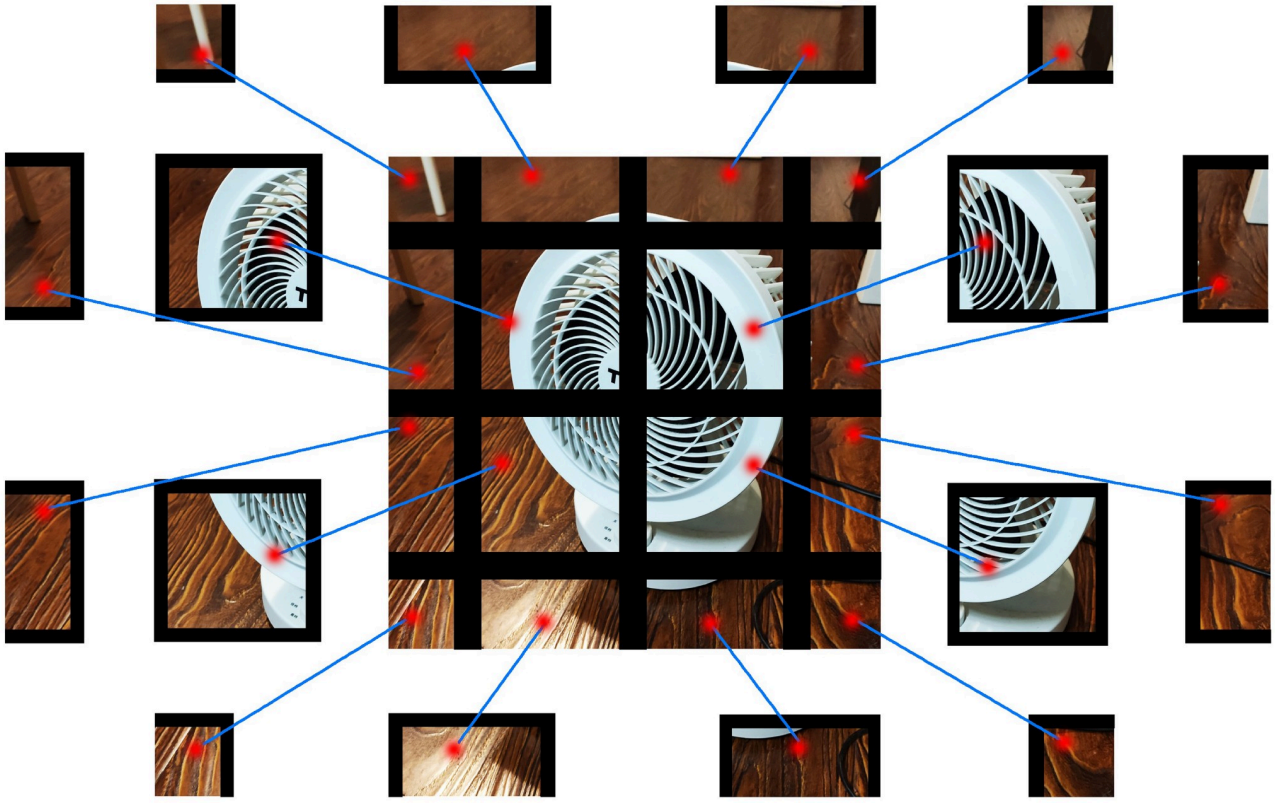

**Fig 1.**

https://doi.org/10.1371/journal.pone.0243613.g001

Similar to the process reported in [13–16], the mask is superposed on the input image only in the training phase; in other words, the image without any data augmentation is put into the network for testing, as shown in Fig 2. Because the network has learned the representations distributed in multiple relevant parts, the segmentation operation is not required during testing.

## Experiment

To verify the validity and utility of our method, the experiments were performed with different computer vision tasks on the PyTorch platform. The performance evaluation standard for the image classification, object detection, and semantic segmentation tasks were top-1 accuracy, mAP, and mIoU, respectively, whose statistical approaches are described in [1, 5, 9, 10]. For demonstrating the performance difference, we selected some of the existing widely used state-of-the-art methods (e.g. Cutout, HaS, AutoAugment, and GridMask) for comparison in each task, all of whose results were cited from the corresponding original papers.

### Image classification

**Imagenet.** We ran our experiment on the ILSVRC-2012 dataset www.image-net.org to verify the performance of the proposed augmentation method. At the time of our experiments, the dataset had 1.4 million images: 1.2 million for training, 0.05 million for validation, and

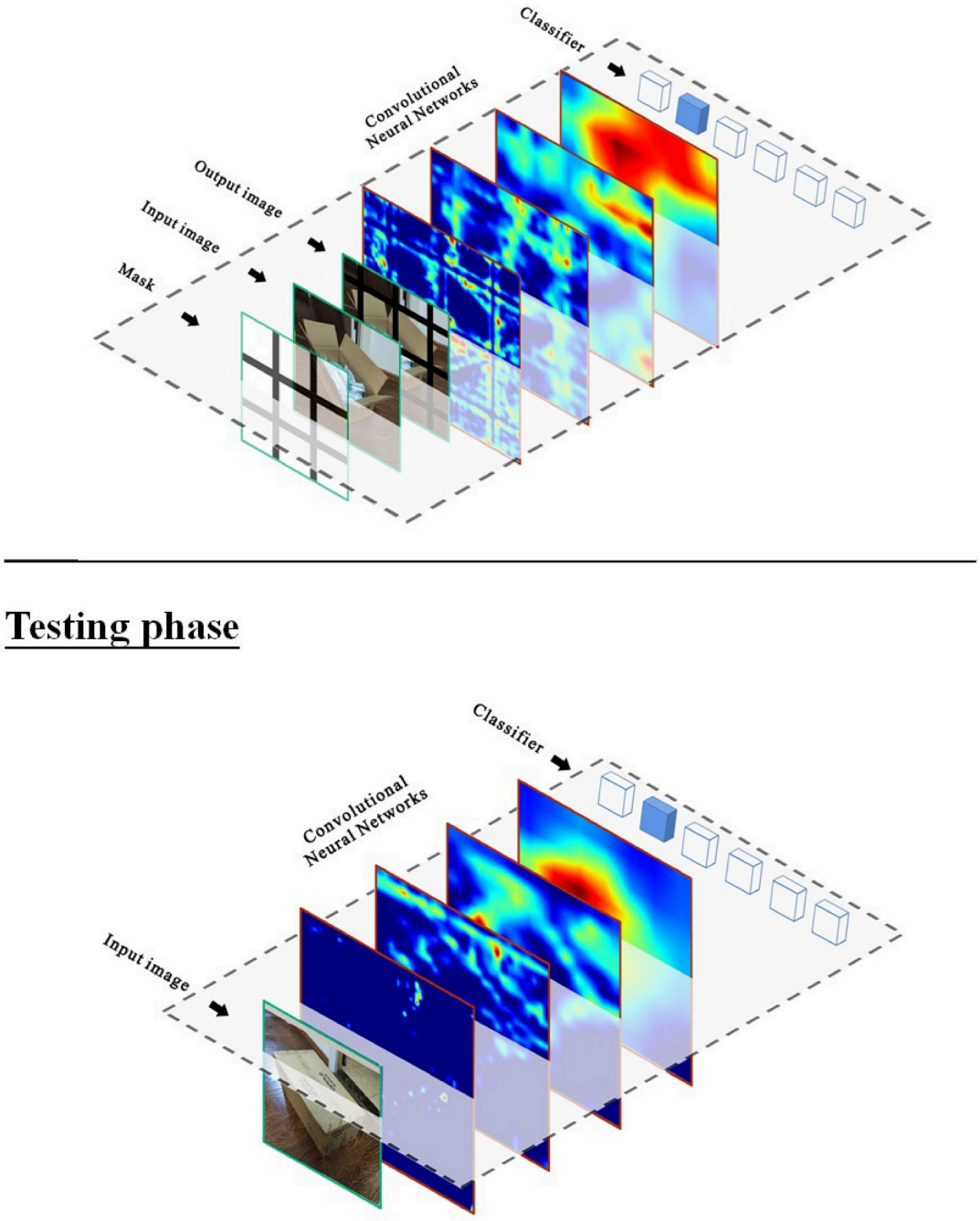


**Fig 2.**

https://doi.org/10.1371/journal.pone.0243613.g002

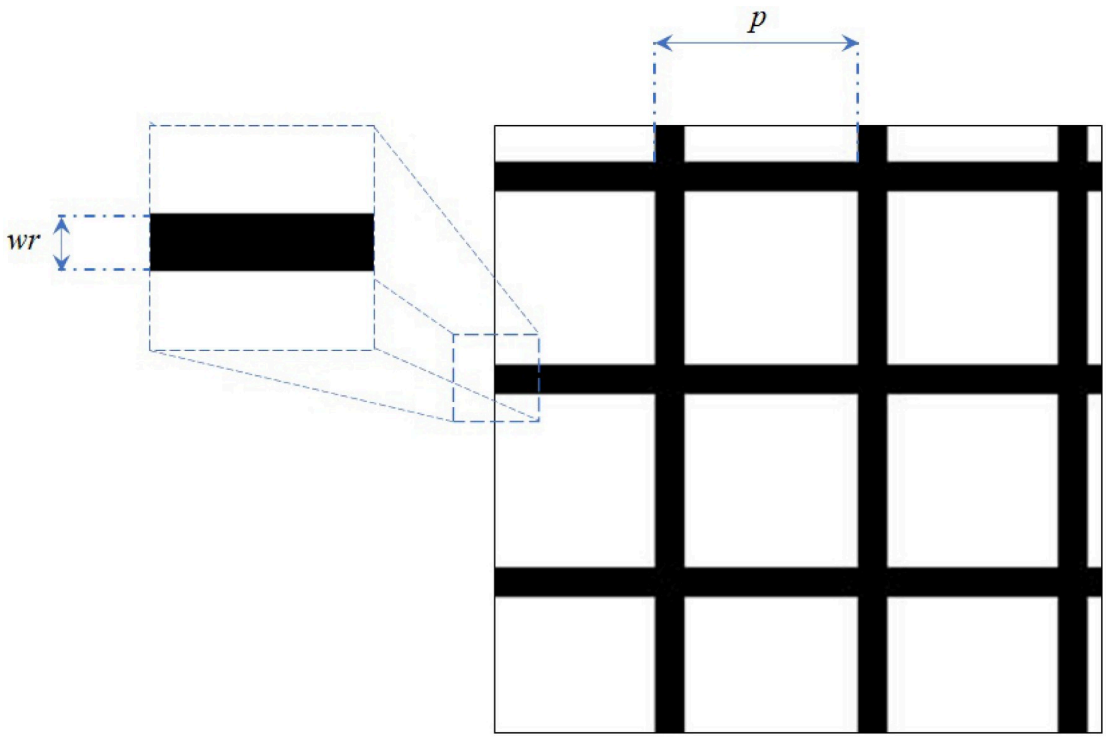


**Fig 3.**

https://doi.org/10.1371/journal.pone.0243613.g003

0.15 million for testing. All of the images were on an average labelled with 1,000 categories for classification. For this dataset, we experimented with the official models of ResNet50 and ResNet101, which had been trained for 300 epochs with a batch size of 256. In the experiment, we used the common practice of baseline augmentation: resize a randomly cropped patch to the target size of 224 × 224 and horizontally flip it with a probability of 0.5.

During the training, we chose the hyperparameters of MeshCut as $p_{max} = 224$, $p_{min} = 81$, and wr = 0.14. The $m_{prob}$ linearly increased from 0 to 0.8 with the training epochs increasing from 0 to 240 and then maintained this value to the end. The learning rate was determined by a MultiStepLR scheduler with an initial value of 0.1, which was reduced by 10% at the 100th, 200th, and 250th epochs. This proposed MeshCut method in our experiment was not accompanied by any other data augmentations, e.g. photometric distortion or information dropping. After all training epochs, the validation loss for the model trained with the MeshCut strategy can converge to a considerably smaller value at a relatively faster rate than that with the baseline augmentation, whereas its training loss is larger value than that with the baseline augmentation. Actually, this is an obvious signal that the MeshCut can enhance the learning difficulty, effectively forcing the network to learn more target features.

Because models after a longer training scheme are usually overfitted, to be fair, we selected the results with the highest validation accuracy in the entire training phase for comparison, as shown in Table 1. The comparison showed that the performance of the proposed MeshCut on the ILSVRC-2012 dataset improved ResNet50 by 1.8% (from 76.5% to 78.3%) and ResNet101 by 2.1% (from 77.8% to 79.9%), which outperformed the other listed augmentation methods. Another noteworthy aspect is the improved results of MeshCut after we incorporated it into AutoAugment, which achieved the state-of-the-art results on the two models.

**Table 1. Experimental results of an image classification task on the ILSVRC-2012 dataset.**

| ILSVRC-2012 | |
|---|---|
| ***Model*** | **Accuracy (%)** |
| *RseNet*50 [19] | 76.5 |
| + *Cutout* [15] | 77.1 |
| + *HaS* [16] | 77.2 |
| + *AutoAugment* [24] | 77.6 |
| + *GridMask* [13] | 77.9 |
| + *MeshCut* | 78.3 |
| + *MeshCut*+ *AutoAugment* | 78.6 |
| *RseNet*101 [20] | 77.8 |
| + *GridMask* [13] | 79.1 |
| + *MeshCut* | 79.9 |
| + *MeshCut*+ *AutoAugment* | 80.1 |

https://doi.org/10.1371/journal.pone.0243613.t001

**CIFAR-10.** The CIFAR-10 dataset www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz contained 50,000 images for training and 10,000 images for testing, all of which were labelled with 10 classes. We used the official ResNet-18 and WideResNet-28-10 models for the experiment. We set the hyperparameters $p_{max}$, $p_{min}$, and wr at 32, 10, and 0.15 respectively.

In term of the baseline augmentation, we first resized the input image to 40 × 40 and randomly cropped a patch with a size of 32 × 32. This experiment was then performed with the same training strategy and statistical method as those discussed in [15] and the Section of Imagenet: the results against other augmentation methods are presented in Table 2.

**Table 2. Experimental results of the image classification task on the CIFAR-10 dataset.**

| CIFAR-10 | |
|---|---|
| ***Model*** | **Accuracy (%)** |
| *RseNet*18 [4] | 95.28 |
| + *Cutout* [15] | 96.25 |
| + *HaS* [16] | 96.10 |
| + *AutoAugment* [24] | 96.07 |
| + *GridMask* [13] | 96.54 |
| + *MeshCut* | 96.79 |
| + *Cutout*+ *AutoAugment* [24] | 96.51 |
| + *GridMask*+ *AutoAugment* [13] | 96.64 |
| + *MeshCut*+ *AutoAugment* | 96.87 |
| *WideResNet* – 28 – 10 [25] | 96.13 |
| + *Cutout* [15] | 97.04 |
| + *HaS* [16] | 96.94 |
| + *AutoAugment* [24] | 97.01 |
| + *GridMask* [13] | 97.24 |
| + *MeshCut* | 97.52 |
| + *Cutout*+ *AutoAugment* [24] | 97.39 |
| + *GridMask*+ *AutoAugment* [13] | 97.48 |
| + *MeshCut*+ *AutoAugment* | 97.64 |

https://doi.org/10.1371/journal.pone.0243613.t002

The experimental results show that the proposed MeshCut strategy could improve upon the accuracies of all baseline models by a large margin and markedly surpassed all the listed information-dropping methods.

### Object detection

To determine the validity and the generalisation ability of MeshCut in an object detection task, we performed an experiment with Faster-RCNN-R50-FPN and Faster-RCNN-X101-FPN on the COCO 2017 dataset cocodataset.org. During the experiment, the models were initialised with the pre-trained weight of ImageNet and then fine-tuned on the COCO dataset. Photometric distortions, geometric transforms, normalised operation, and MeshCut were performed in this sequence. To determine the ability of MeshCut, we trained the models with the same hyperparameters as those described in [8], with 1×, 2×, and 4× training epochs. The other hyperparameters for MeshCut were the same as those given in the ImageNet section. The experimental results of MeshCut with various epochs are summarised in Table 3. In terms of the Faster-RCNN-R50-FPN, the results of mAP increased from 37.4% (baseline) to 38.6% (+1.2%) for 2× epochs, and from 37.4% (baseline) to 39.7% (+2.3%) for 4× epochs. For Faster-RCNN-X101-FPN, the results of mAP increased from 41.2% (baseline) to 42.9% (+1.7%) for 2× epochs. Compared with GridMask, our strategy could also achieve the best results in this task.

### Semantic segmentation

To determine the capacity and the universality of MeshCut in a semantic segmentation task, we performed an experiment with the PSPNet model on the Cityscapes dataset [www.cityscapes-dataset.com]. We trained the model with the same hyperparameters as those suggested in [10], except for the longer (2×) training epochs. The above PSPNet model was also initialised with the pre-trained weights of ImageNet and then fine-tuned on the Cityscapes dataset www.cityscapes-dataset.com. The hyperparameters on MeshCut were the same as those given in the ImageNet section.

Although the mIoU of the PSPNet model could be improved significantly by introducing the GridMask, our MeshCut still achieved better results, as shown in Table 4.

## Discussion

In this section, we present an analysis to show the influences of various hyperparameters and the changes in the class activation mappings (CAMs) [15]. The related experiments were

**Table 3. Experimental results of the object detection task on the COCO dataset.**

| COCO2017 | | | |
|---|---|---|---|
| ***Model*** | **mAP (%)** | **AP50 (%)** | **AP75 (%)** |
| *Faster − RCNN − R50 − FPN* [8] | 37.4 | 58.7 | 40.5 |
| +*gridmask*(2×) [13] | 38.3 | 60.4 | 41.7 |
| +*MeshCut*(2×) | 38.6 | 60.5 | 41.9 |
| +*gridmask*(4×) [13] | 39.2 | 60.8 | 42.2 |
| +*MeshCut*(4×) | 39.7 | 61.2 | 42.5 |
| *Faster − RCNN − X101 − FPN* [8] | 41.3 | 63.5 | 44.4 |
| +*gridmask*(2×) [13] | 42.6 | 65.0 | 46.5 |
| +*MeshCut*(2×) | 42.9 | 65.3 | 46.9 |

https://doi.org/10.1371/journal.pone.0243613.t003

**Table 4. Experimental results of the semantic segmentation task on the Cityscapes dataset.**

| Cityscapes | |
|---|---|
| ***Model*** | **mIoU (%)** |
| *PSPNet*50 [10] | 77.2 |
| + *gridmask* [13] | 78.1 |
| + *MeshCut* | 78.5 |
| *PSPNet*101 [10] | 78.3 |
| + *gridmask* [13] | 79.0 |
| + *MeshCut* | 79.3 |

https://doi.org/10.1371/journal.pone.0243613.t004

conducted using the same training and validation strategies as those discussed in the ImageNet and CIFAR-10 sections.

## Hyperparameter p

The line spacing p determines the size of the grid. To explore its impact, we selected ResNet18 to experiment with various ranges of p on CIFAR-10. The experiment results are presented in Table 5. For the same maximum bound ($p_{max}$), the performance of MeshCut was better when p was distributed in a relatively larger interval, showing that a greater diversity of p could effectively improve the robustness of the network. In contrast, for the same distribution range, the performance became poor in the case of a small $p_{min}$. The phenomenon that a smaller p was helpful to avoid segmentation failures did not seem to make sense, but it could be explained by the crash of feature extractions caused by the overfragmented segmentation. To effectively avoid this problem, we recommend a set of theoretical formulas as guideline to speed the hyperparameter tunning, as following:

$$\mathrm{p}_{\max} = max(size(image)), \tag{4}$$

$$\mathrm{p}_{\min} = max(size(target))/2, \tag{5}$$

where the size(.) represents the statistic operator for the number of pixels in horizontal and vertical directions, the max(.) represents the operator for achieving the maximum value from the array. For the Imagenet and CIFAR-10, the majority of the target sizes are up to 60 90% of the image size, hence the optimal pmin in our experiment is within a range from 1/2 to 1/3 of image size.

## Hyperparameter wr

The hyperparameter wr determines the information retention ratio for an input image, which is important for the proposed algorithm to achieve good results, because artefacts may be

**Table 5. The results of various ranges of p.**

| Hyperparameter p | |
|---|---|
| ***Range of p*** | **Accuracy (%)** |
| [8, 24] | 95.37 |
| [10, 26] | 96.48 |
| [16, 32] | 96.56 |
| [13, 32] | 96.67 |
| [10, 32] | 96.79 |

https://doi.org/10.1371/journal.pone.0243613.t005

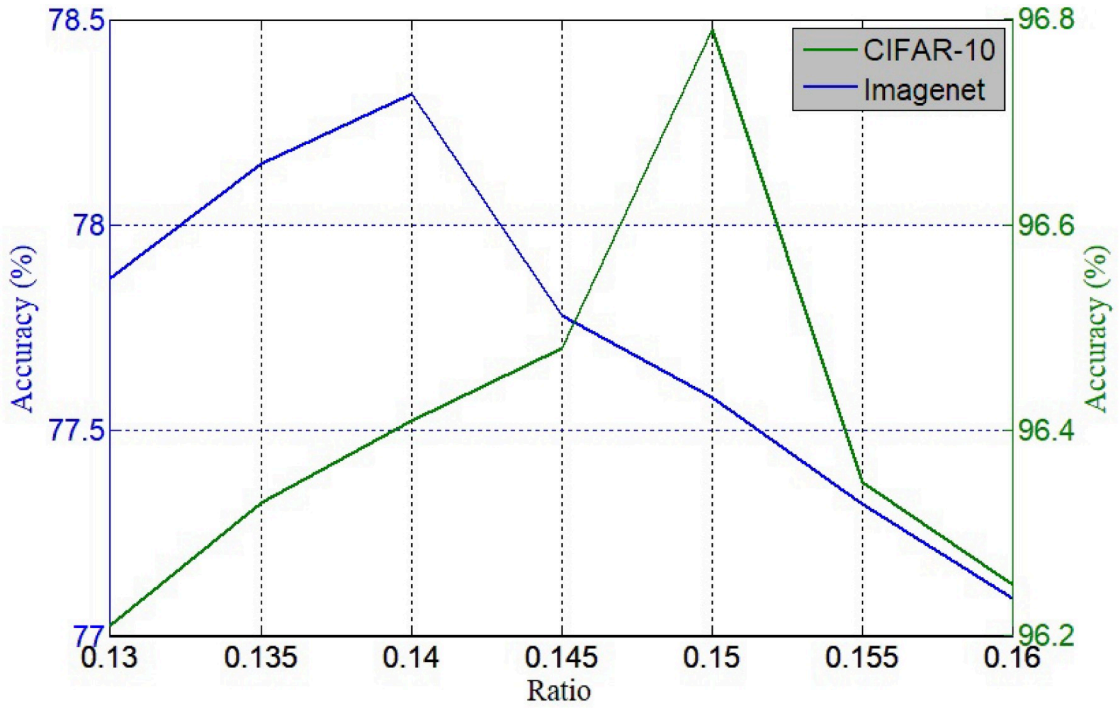


**Fig 4.**

https://doi.org/10.1371/journal.pone.0243613.g004

unexpectedly introduced into a network in the case of a very small wr, whereas feature information may be excessively lost in the case of a very large wr. In practical use, wr is a constant, and its influence on the experimental results is shown in Fig 4.

Fig 4 shows the accuracy curve of ResNet50 trained on ImageNet. The figure shows that the performance of the proposed strategy was relatively poor in the case of a small wr. Under this condition, the capacity of MeshCut was limited, and the algorithm could not efficiently solve the overfitting. With an increase in wr, the accuracy improved and reached a maximum of 78.1% when the wr was at 0.14. Beyond this, there was a monotonic decrease in accuracy, meaning that underfitting began to appear. The same effect occurred on the curve of CIFAR-10 with ResNet18, except for the optimal value of wr, as shown in Fig 4. This very small difference made sense: more information needs to be reserved for perception in the case of a more complex dataset.

Actually, the wr determines the effective action depth of segmentation operation in the network, e.g. the larger segmentation gap still remains valid in a deeper layer after several down-sampling operations. Generally, the segmentation is expected to impact on the mid-level features, because the segmentation for the low-level features cannot efficiently implement the semantic segmentation while that for the high-level features may cause the problem of information loss. Herein, we also recommend a theoretical formula as guideline:

$$\mathrm{wr} = [n_{kernel} * size(image)/size(mid - level)]/p_{min}, \tag{6}$$

where $n_{kernel}$ is the size of convolution kernel, the size(mid-level) represents the input size of the down-sampling layer before the deepest layer on which the segmentation remains effect. For the Imagenet, the deepest effectively-segmented layer is selected the conv3-3 of the

**Table 6. Results with two strategies for different values of $m_{prob}$.**

| $m_{prob}$ | |
|---|---|
| ***Model*** | **Accuracy (%)** |
| *MeshCut+ fixed* $m_{prob}$ | 96.43 |
| *MeshCut+ increasing* $m_{prob}$ | 96.79 |

https://doi.org/10.1371/journal.pone.0243613.t006

ResNet50 model, hence $n_{kernel} = 3$, size(mid-level) = 56. After calculation, the wr is within a range around 0.14. The same scheme can be applied for the CIFAR-10, and the similar results can be achieved.

## Scheme to use MeshCut

As for $m_{prob}$, there exist two ways to apply MeshCut in practice: 1) training with a fixed $m_{prob}$ and 2) training with increasing $m_{prob}$. To determine a better-suited implementation, we selected ResNet18 to experiment on CIFAR-10 with two different schemes. For the first strategy, we set $m_{prob}$ at 0.8 during the entire training phase. For the second strategy, we performed the experiment with the same strategy as that described in Section of CIFAR-10. The results showed that the second strategy led to a better result, as shown in Table 6.

## Loss curves

The loss curve is a critical indicator that reflects the affection of MeshCut on the training process. Considering this, we selected ResNet18 to experiment with baseline and MeshCut data augmentation schemes on CIFAR-10, whose loss curves in both training and validation phases are shown in Fig 5. Seen from it, the training loss for the model with baseline augment decreased to 0.002 after several epochs, while the validation loss only converged to 0.14, then the further decline became increasingly difficult. Comparing with this model, the training loss of model trained MeshCut strategy fell to 0.03 at a relatively slower rate, whereas its validation loss could dip below a much smaller value of 0.10. This is an obvious signal that the MeshCut could enhance the learning difficulty, effectively forcing the network to learn more target features.

## CAMs

The key idea of MeshCut is to generate information occlusion with a mesh mask, with the purpose of forcing the network to learn more spatially distributed representations instead of

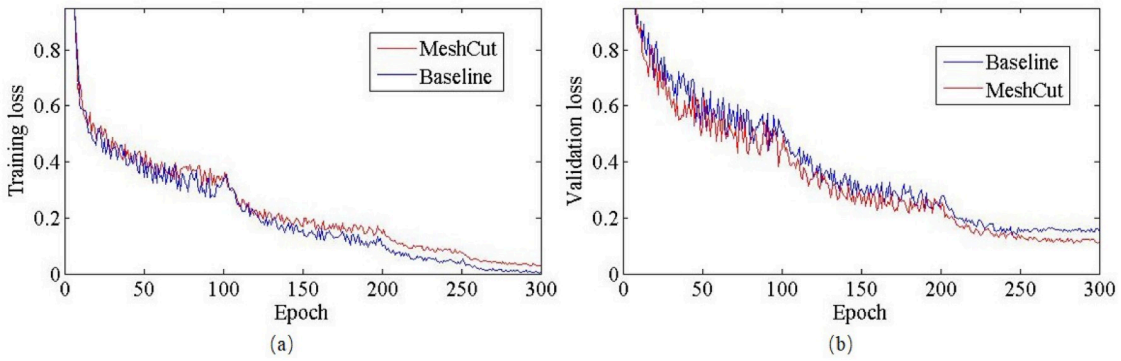


**Fig 5.**

https://doi.org/10.1371/journal.pone.0243613.g005

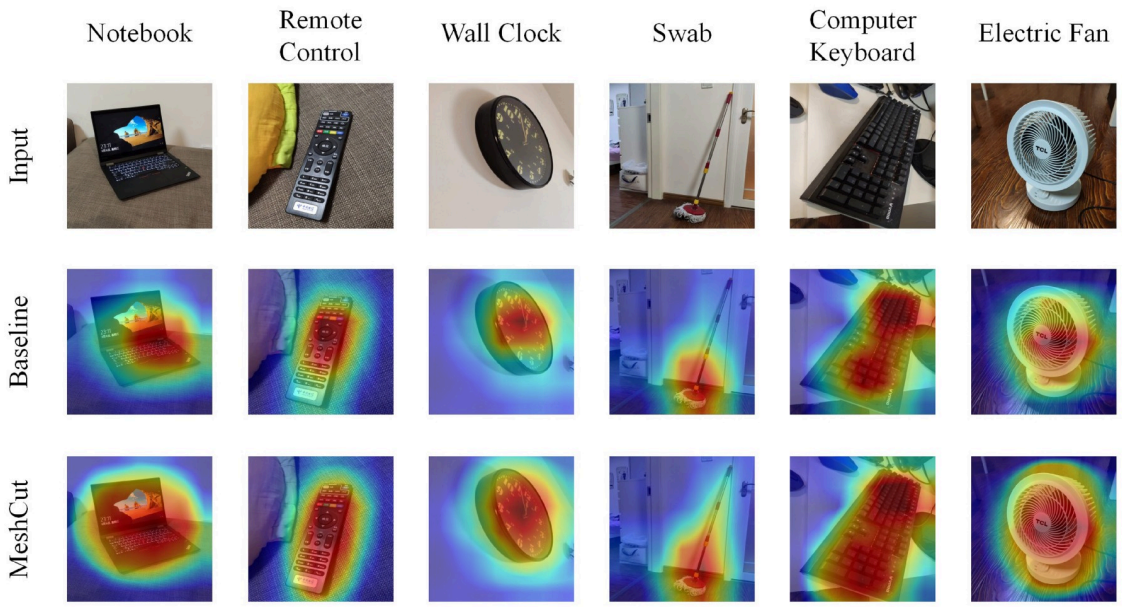


**Fig 6.**

https://doi.org/10.1371/journal.pone.0243613.g006

focusing on only one discriminative feature. Therefore, the features learned by the network could directly respond to the efficiency of MeshCut. We introduced the CAMs for the conv5-3 of the ResNet50 model trained on ImageNet for the analysis. The comparison CAMs for various augmentation methods are shown in Fig 6. As a result, MeshCut trended to focus on larger spatially distributed regions, while the baseline augmentation tended to concentrate on a small region.

## Conclusion

We proposed a novel information-segregation-based augmentation strategy to solve the problem of overfitting for CNNs. By virtue of its mesh structure, the proposed MeshCut achieved state-of-the-art experimental results in various tasks and models. Extensive experiments analysed the influences of various hyperparameters on its performance. Moreover, the performance of such a strategy could be improved further by incorporation into other augmentation strategies, e.g. AutoAugment, which could make MeshCut a promising baseline strategy for data augmentation algorithms.

## Author Contributions

**Conceptualization:** Wei Jiang.

**Data curation:** Kai Zhang.

**Funding acquisition:** Miao Yu.

**Software:** Nan Wang.

**Writing – original draft:** Wei Jiang.

**Writing – review & editing:** Nan Wang.

## References

1. Krizhevsky A., Sutskever I., and Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks. Communications Of the Acm. 2017 Jun; 60:84–90.
2. Simonyan K. and Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Computer Science. 2014.
3. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. Rethinking the Inception Architecture for Computer Vision. IEEE Conference on Computer Vision And Pattern Recognition. 2016:2818-2826.
4. K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision And Pattern Recognition. 2016:770-778.
5. R. Girshick. Fast R-CNN. IEEE Conference on Computer Vision And Pattern Recognition. 2015:1440-1448.
6. R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE Conference on Computer Vision And Pattern Recognition. 2014:580-587.
7. K. M. He, G. Gkioxari, P. Dollar, R. Girshick. Rethinking the Inception Architecture for Computer Vision. IEEE Conference on Computer Vision And Pattern Recognition. 2017:2980-2988.
8. S. Q. Ren, K. M. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Conference on Computer Vision And Pattern Recognition. 2017 Jun;39:1137-1149.
9. J. Long, E. Shelhamer, T. Darrell. Fully Convolutional Networks for Semantic Segmentation. IEEE Conference on Computer Vision And Pattern Recognition. 2015:3431-3440.
10. H. S. Zhao, J. P. Shi, X. J. Qi, X. G. Wang, J. Y. Jia. Pyramid Scene Parsing Network. IEEE Conference on Computer Vision And Pattern Recognition. 2017:6230-6239.
11. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis And Machine Intelligence. 2018 Apr,40:834-848.
12. C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, et al. Going Deeper with Convolutions. IEEE Conference on Computer Vision And Pattern Recognition. 2015:1-9.
13. Zhou B., Khosla A., Lapedriza A., Oliva A., Torralba A. Learning Deep Features for Discriminative Localization. IEEE Conference on Computer Vision And Pattern Recognition. 2016;2921–2929.
14. Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random Erasing Data Augmentation. arXiv. 2017.
15. T. DeVries and G. W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. arXiv. 2017.
16. K. K. Singh, H. Yu, A. Sarmasi, G. Pradeep, and Y. J. Lee. Hide-and-Seek: A Data Augmentation Technique for Weakly-Supervised Localization and Beyond. arXiv. 2018.
17. Li W., Zeiler M. D., Zhang S., Lecun Y., and Fergus R. Regularization of Neural Networks using DropConnect. International Conference on Machine Learning. 2013.
18. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., and Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal Of Machine Learning Research. 2014 Jun, 15:1929–1958.
19. Ghiasi G., Lin T. Y., and Le Q. V. DropBlock: A regularization method for convolutional networks. Neural Information Processing Systems. 2018; 31.
20. H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond Empirical Risk Minimization. arXiv. 2017.
21. S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. arXiv. 2019.
22. Shiyao Wang, Yucong Zhou, Junjie Yan, and Zhidong Deng. Fully Motion-Aware Network for Video Object Detection. European Conference on Computer Vision. 2018;1-16.
23. Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Amir Hussain. Convolutional MKL Based Multimodal Emotion Recognition and Sentiment Analysis. International Conference on Data Mining. 2016;439-448.
24. E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. AutoAugment: Learning Augmentation Strategies From Data. IEEE Conference on Computer Vision And Pattern Recognition. 2019;113-123.
25. Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. arXiv. 2017.