Corrdrop: Correlation Based Dropout for Convolutional Neural Networks

Conference Paper · May 2020

DOI: 10.1109/ICASSP40776.2020.9053605

Yuyuan Zeng
Tsinghua University

Tao Dai
Tsinghua University

CORRDROP: CORRELATION BASED DROPOUT FOR CONVOLUTIONAL NEURAL NETWORKS

Yuyuan Zeng[⋆,†] *Tao Dai*[⋆,†,‡] *Shu-Tao Xia*[⋆,†]

[⋆]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[†] PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China
zengyy19@mails.tsinghua.edu.cn, daitao.edu@gmail.com, xiast@sz.tsinghua.edu.cn

ABSTRACT

Convolutional neural networks (CNNs) can be easily over-fitted when they are over-parametered. The popular dropout that drops feature units randomly can't always work well for CNNs, due to the problem of under-dropping. To eliminate this problem, some structural dropout methods such as SpatialDropout, Cutout and DropBlock have been proposed. However, these methods that drop feature units in continuous regions randomly, may have the risk of over-dropping, thus leading to degradation of performance. To address these issues, we propose a novel structural dropout method, Correlation based Dropout (CorrDrop), to regularize CNNs by dropping feature units based on feature correlation, which reflects the discriminative information in feature maps. Specifically, the proposed method first obtains correlation map based on the activation in the feature maps, and then adaptively masks out those regions with small average correlation. Thus, the proposed method can regularize CNNs well by discarding part of contextual regions. Extensive experiments on image classification demonstrate the superiority of our method compared with other counterparts.

Index Terms— Over-fitting, Regularization, Dropout, Convolutional Neural Networks

1. INTRODUCTION

Convolutional neural networks (CNNs) have been widely and successfully used in various computer vision tasks [1, 2, 3]. In recent years, various architectures of deep CNNs with powerful representations, such as ResNet [4], InceptionNet [5], and DenseNet [6], are proposed to improve the performance of CNNs. It is known that network parameters increase quickly with the growth of layers. Thus, it is easy to raise the problem of over-fitting for deep CNNs especially on small datasets. Therefore, it is worthy to develop regularization methods to relieve the problem of over-fitting for CNNs.

Early proposed regularization methods, such as weight decay [7], early stopping [8], data augmentation [9, 10] and dropout [11] have been widely used in deep neural networks. Among them, dropout has achieved immense success for fully connected networks for its powerful regularization ability. However, recent study shows that the traditional dropout is less effective for CNNs due to the problem of under-dropping, since the spatially correlated features still allow dropped information to flow through the network [12].

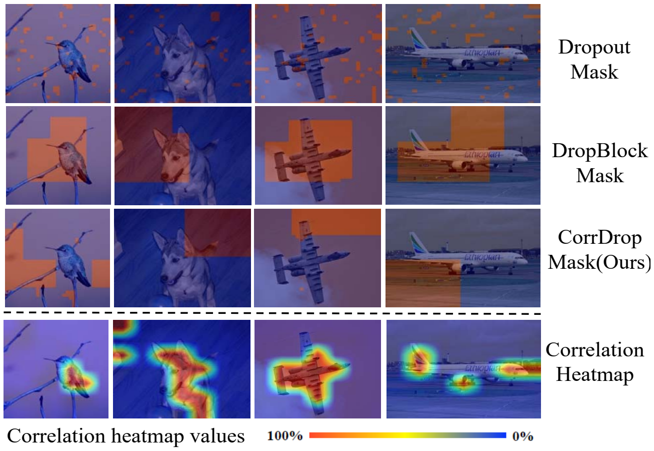


Fig. 1. Masks of Dropout [11], DropBlock [12] and our CorrDrop and the correlation heatmap produced by CorrDrop. The red regions denote the regions to be masked. Compared with Dropout and DropBlock, CorrDrop takes the discriminative information into consideration and drops feature units adaptively to alleviate the under- and over-dropping problems.

To make dropout more effective for CNNs, some structural dropout methods have recently been proposed, such as SpatialDropout [13], Cutout [14], DropBlock [12] by dropping entire channels or square of regions in the input/feature space. However, these structural dropout methods confront the risk of over-dropping, since they drop units in continuous regions randomly, perhaps discarding the whole of discriminative regions in the input/feature maps, thus leading to degradation of performance. As shown in Fig. 1, traditional dropout [11] and the recently proposed DropBlock [12] have the risk of under-dropping and over-dropping by dropping feature units randomly.

To address above issues, motivated by the observations that discriminative regions of an object would have higher feature correlations (see the last row in Fig. 1), we propose a novel and effective structual dropout: CorrDrop, which drops feature units according to the feature correlation. In order to obtain a better regularization effect, it is more appropriate to drop the feature units adaptively based on the discriminative information. To this end, we first compute the feature correlation map, and then adaptively mask out those regions with less discriminative information, i.e., regions with small feature correlation. As shown in Fig. 1, compared with Dropout and DropBlock that result in under- and over-dropping, our CorrDrop

[‡] Corresponding author: Tao Dai

This work is supported in part by the National Key Research and Development Program of China under Grant 2018YFB1800204, the National Natural Science Foundation of China under Grant 61771273, the China Postdoctoral Science Foundation under Grant 2019M660645, the RD Program of Shenzhen under Grant JCYJ20180508152204044, and the research fund of PCL Future Regional Network Facilities for Large-scale Experiments and Applications (PCL2018KP001).

produces adaptive masks by discarding part of contextual semantic information, thus making the network learn more compact representations. Extensive experiments demonstrate that CorrDrop outperforms Dropout [11] and DropBlock [12] by precisely dropping unimportant features which encourages the network to learn meaningful representation.

2. RELATED WORKS

Regularization in Deep Learning. Deep neural networks with huge amount of parameters can be easily over-fitted, hindering the generalization of the models. To solve the problem, many regularization methods [7, 8, 11, 15, 16, 17] have been proposed in the past few years. Among them, dropout [11] has been shown to significantly improve the performance of deep neural networks for years. However, recent research [12] indicate that the traditional dropout suffers from under-dropping problem when used in CNNs, since features are locally correlated in CNNs. Later, a plenty of dropout variants such as SpatilDropout [13], Cutout [14] and DropBlock [12] are proposed. However, these methods may have the risk of over-dropping by discarding features randomly with equal dropout probability. Instead, our proposed method CorrDrop alleviates these problems by dropping feature units adaptively considering feature correlation.

Attention Mechanism. Inspired by the phenomenon that humans tend to focus on the discriminative parts of the images, attention mechanism has been widely used in various fields such as machine translation [18], image classification [19], transfer learning [20], wealy supervised object localization [21] and etc. which greatly improves the performance. Attention mechanism considers the correlation of the features. When the query comes in, it focuses more on the important data. Similarly, our method takes the feature correlation into account, however, CorrDrop generates the attention map based on the correlation calculation introduced in Sec. 3.1 which does not add any additional trainable parameters and greatly relieves the parameter overheads. Besides, CorrDrop combines attention mechanism with dropout which improve the classification performance as a new variant of regularization.

3. METHODOLOGY

Existing dropout-based methods undergo the risk of under- or over-dropping. To render dropout more effective for CNNs, we propose a simple but effective structural dropout: CorrDrop, which drops the feature units adaptively based on the discriminative information. The pipeline of our algorithm is shown in Fig. 2. Specifically, we take feature correlation into account and assign each unit with adaptive dropout probability according to their correlation score. In this section, we firstly describe the calculation of feature correlation based on feature orthogonality. Then we illustrate the strategy of correlation based dropout. Finally, following the previous work DropBlock [12], to regularize CNNs better, we further generate structural dropout mask by dropping the square feature regions.

3.1. Feature Correlation Calculation

In spatial dimension, we suppose that highly correlated units construct the discriminative parts in the feature maps, which should be kept with higher probability. The metric based on the feature orthogonality is demonstrated to be a satisfactory way to represent feature correlation [22]. Given the feature maps of the intermediate l-th layer as $A^{(l)} = [a_1^{(l)}, ..., a_N^{(l)}]^T \in \mathbb{R}^{N \times C}$, where $N = H \times W$ is the number of units in a feature map, C is the number of channels, H and W are the height and width of the feature map respectively. Each row $a_i^{(l)} \in \mathbb{R}^{\mathbb{C}}$ represents the feature vector of a unit. The correlation calculation can be described as below,

$$\widehat{A}^{(l)} = \frac{A^{(l)}}{\|A^{(l)}\|}, \tag{1}$$

$$P^{(l)} = |\widehat{A}^{(l)} \times \widehat{A}^{(l)T} - I|, \tag{2}$$

$$F_i^{(l)} = \frac{\sum_N P_i^{(l)}}{N} \tag{3}$$

where I is an identity matrix with size $N \times N$. We first normalize each row of $A^{(l)}$ and compute the correlation scores based on the feature orthogonality. $P^{(l)}$ is a matrix of size $N \times N$ and $P_i^{(l)}$ denotes i-th row of $P^{(l)}$. Off-diagonal elements of a row of $P^{(l)}$ for a single unit denote projection of all the other units in the same feature map. The mean of each row denotes the correlation score of each unit. The higher value of $F_i^{(l)}$ indicates that this unit is highly correlated with others.

3.2. Correlation Based Dropout

To drop units adaptively based on the feature correlation, we assign the dropout probability to each unit according to the values in F. Generally, the higher value of $F_i^{(l)}$ is, the smaller dropout probability we have,

$$\gamma_{i,j}^{(l)} = 1 - \frac{F_{i,j}^{(l)} - F_{min}^{(l)}}{F_{max}^{(l)} - F_{min}^{(l)}}. \tag{4}$$

To obtain the dropout probability, we normalize the correlation score of each unit to ensure that $\gamma_{i,j}^{(l)} \in (0, 1)$. The dropout mask $M^{(l)} \in \mathbb{R}^{H \times W}$ is sampled from Bernoulli distribution with correlation based dropout probability γ,

$$M_{i,j}^{(l)} = Bernoulli(1 - \gamma_{i,j}^{(l)}). \tag{5}$$

Empirically, similar to other dropout variants, a hyper-parameter the dropout probability p is introduced to ensure the algorithm will not drop too many units. With the correlation based dropout mask $M^{(l)}$, we adjust the retain probability and generate another mask $B^{(l)} \in \mathbb{R}^{H \times W}$. The units are dropped when the corresponding values in both two masks are 0 and the final dropout mask $S^{(l)} \in \mathbb{R}^{H \times W}$ is obtained.

$$\widehat{p} = \frac{p \times numel(M^{(l)})}{numel(M^{(l)}) - sum(M^{(l)})}, \tag{6}$$

$$B_{i,j}^{(l)} = Bernoulli(1 - \widehat{p}), \tag{7}$$

$$S_{i,j}^{(l)} = \begin{cases} 0 & M_{i,j}^{(l)} = 0 \; and \; B_{i,j}^{(l)} = 0 \\ 1 & otherwise \end{cases}, \tag{8}$$

$$\widetilde{A}^{(l)} = S^{(l)} \odot A^{(l)}, \tag{9}$$

where $numel(M^{(l)})$ counts the number of units in $M^{(l)}$, $sum(M^{(l)})$ counts the number of units where the value is 1, and $\odot$ represents the point-wise multiplication operation.

As the features are locally correlated in CNNs, it is less effective to drop a single unit in the feature map [12]. Following the previous work DropBlock [12] which drops continuous regions in the feature map, we further consider the correlation of each local area and drop blocks of units. To obtain a structural mask, we firstly gather the local information in the feature map by local average pooling

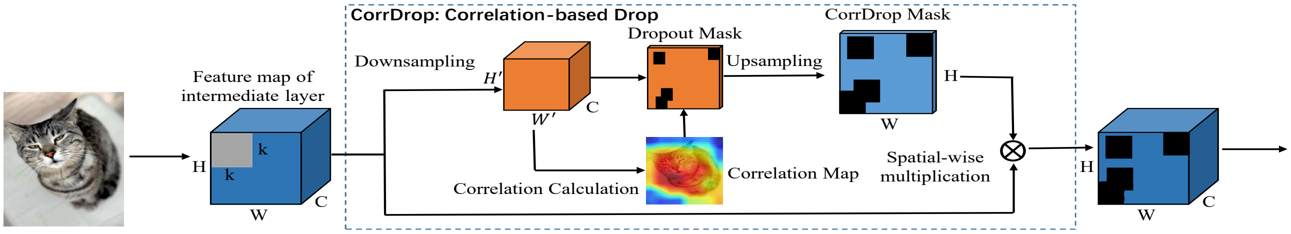


Fig. 2. The pipeline of CorrDrop. 1) Downsample the feature maps from previous layers by spatial-wise local average pooling with kernel size and stride of k for local features gathering and dimensional reduction. 2) A correlation map is calculated based on the feature orthogonality introduced in Section 3.1, with which the dropout mask is sampled from Bernoulli distribution with adaptive dropout probability. 3) The structural CorrDrop mask is generated by nearest neighbor upsampling, thus blocks with size of $k \times k$ near each zero entry in dropout mask are dropped. 4) The CorrDrop mask is multiplied to each channel of the original feature maps to drop features.

and meanwhile reduce the dimension of the feature map for speeding up the correlation calculation; and then produce the correlation based dropout mask as illustrated before; finally generate the structural mask by nearest neighbour upsampling and drop square regions of units. The pipeline is shown in Fig. 2. In this manner, we calculate the feature correlation based on the local information and drop square of regions with small average correlation.

The process can be mathematically described as below. When setting the block size to k, we perform local average pooling on the feature map with kernel size of k and stride of k. Specifically, we scan each block with size of $k \times k$ from left to right, top to bottom in each feature map and compute the mean of activation values in each block, which can be described as

$$a^{(l)'}_{i',j'} = \frac{\sum_{k_1=-\frac{k}{2}}^{\frac{k}{2}} \sum_{k_2=-\frac{k}{2}}^{\frac{k}{2}} a^{(l)}_{i+k_1,j+k_2}}{k^2}. \qquad (10)$$

The resulted feature map is $A^{(l)'} \in \mathbb{R}^{N' \times C}$, where $N' = H' \times W'$, $H' = ceil(\frac{H}{k})$, $W' = ceil(\frac{W}{k})$. Moreover, in the implementation of nearest neighbor upsampling, every zero entry in the dropout mask will be expanded by k^2, we need to adjust the dropout probability p in order to keep every unit with dropout probability of p. Specifically, the adjusted dropout probability p' can be computed by

$$p' = \frac{p}{k^2} \frac{H \times W}{H' \times W'}. \qquad (11)$$

Thus, we use the adjusted dropout probability p' to sample the initial binary mask in Equ. (6). With the dropout mask $S^{(l)'} \in \mathbb{R}^{H' \times W'}$ produced based on the downsampled feature map $A^{(l)'}$, we upsample $S^{(l)'}$ with nearest neighbor upsampling and drop a block of size $k \times k$ near each zero entry in $S^{(l)'}$ and produce the structural CorrDrop mask $S^{(l)} \in \mathbb{R}^{H \times W}$. Finally the CorrDrop mask is multiplied to each channel of the original feature maps $A^{(l)}$ and masks out part of feature regions.

4. EXPERIMENTS

To evaluate the effectiveness of our CorrDrop method, we compare it with other state-of-the-art dropout-based methods [11, 13, 12, 14] on image classification with CIFAR-10 and CIFAR-100 [23] datasets. In addition, we also conduct experiments on different architectures, different choices of hyper-parameters and visualization of class activation map.

4.1. Experimental Settings

For image classification, we normalize the datasets with per-channel mean and deviation. Standard data augmentation schemes like flipping, random cropping are also incorporated. We report the highest validation accuracy following common practice. Unless otherwise specified, all experiments are based on ResNet20 [4] using the official PyTorch implementation. The defalut setting of batch size is 128, the optimizer is SGD with momentum of 0.9 [24], the total training epochs is 200 and the initial learning rate is 0.1 and is decayed by the factor of 1e-1 at 0.4, 0.6, 0.8 ratio of total epochs. Following [12], we gradually increase the value of dropout probability p with linear scheduler.

4.2. Classification on CIFAR-10 and CIFAR-100

In this section, we compare the regularization effect of our proposed CorrDrop with other state-of-the-art dropout-based methods. The experimental results are shown in Table 1 and CorrDrop yields best performace. The regularization layers are added after each convolution group in ResNet20. For CorrDrop and DropBlock, we use block size of 9, 7, 5 for three convolutional blocks respectively. CorrDrop outperforms DropBlock consistently. This is mainly because DropBlock undergoes the risk of over-dropping by dropping square feature regions randomly, while ours alleviate this problem by adaptively dropping feature regions based on discriminative information. Moreover, Cutout also works well to regularize CNNs by dropping units in the input/image space as a data augmentation technique. When combining Cutout with our CorrDrop, the test accuracy of ResNet20 on both datasets can be further improved. The test accuracy on CIFAR-10 and CIFAR-100 with different regularization methods during training are displayed in Fig. 3. All of observations demonstrate the effectiveness of our method.

Analysis of computation and parameters overhead. The computation overhead of CorrDrop only comes from the calculation of correlation map as introduced in Sec. 3.1. Due to the use of a local downsampling operation, the size of the feature map is reduced. The number of FLOPs in the training process only increase by 0.003% compared to baseline. While in the test process, CorrDrop is closed like Dropout, so the time complexity would not increase in testing process.

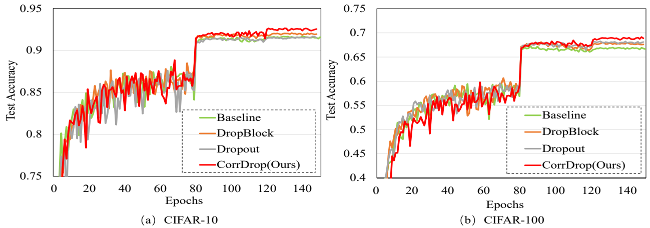
In addition, the calculation of correlation map of CorrDrop based on the feature orthogonality which does not add any additional trainable parameters. Thus when compared to baseline model, model regularized with CorrDrop have the same number of parameters.

4.3. Regularization on Different Architectures

To demonstrate that our method is applicable to different architectures, we conduct image classification on CIFAR10 with different architectures such as VGG16 [25], ResNet110 [4], DenseNet [6]

Table 1. Classification accuracy of ResNet20(Top-1%) on CIFAR-10 and CIFAR-100 with different regularization methods. The value of p is the best parameter decided by grid search.

Methods	CIFAR-10	CIFAR-100
No Regularization	91.79	67.31
Dropout($p = 0.15$)	92.20	68.11
Spatial Dropout($p = 0.1$)	92.01	68.09
DropBlock($p = 0.2$)	92.37	68.11
CorrDrop(Ours)($p = 0.1$)	**92.57**	**68.88**
Cutout	92.49	69.01
Cutout + CorrDrop(Ours)($p = 0.03$)	**92.87**	**69.65**



(a) CIFAR-10 (b) CIFAR-100

Fig. 3. Test accuracy on CIFAR-10(a) and CIFAR-100(b) with different regularization methods during training.

and Wide Residual Network [26]. We add the regularization layer after the first convolution group with block size of 9. Experimental results are shown in Table 2. For deep CNNs, dropout can not effectively regularize the models. Compared with DropBlock, CorrDrop can further improve the performance due to correlation based dropout. The consistent improvement when regularized with CorrDrop demonstrates that our method is practical for training deep CNNs.

Table 2. Classification accuracy(Top-1%) on CIFAR-10 with different architectures regularized by different methods.

Models	Baseline	Dropout	DropBlock	CorrDrop (Ours)
VGG16_BN	93.83	93.57	93.35	**94.09**
ResNet110	93.61	93.65	94.15	**94.38**
DenseNet	95.32	95.31	95.55	**95.78**
WRN-28-10	95.98	95.96	96.38	**96.50**

4.4. Analysis of Hyper-parameters Choice

In this section, we test the sensitiveness of our algorithm to different choices of hyper-parameters: the dropout probability p and block size k. It is worth to mention that compared to DropBlock, our proposed method do not introduce any additional hyper-parameters.

Choice of dropout probability In Fig. 4(a), we display the test accuracy on CIFAR-10 with Dropout, DropBlock and CorrDrop when applying different dropout probability. For fair comparison, the regularization layers are added after the first convolution group in ResNet20. For DropBlock and CorrDrop, the block size is set to 9. Empirically, CorrDrop gains the best performance with the dropout probability of 0.15. Compared with the other methods, CorrDrop achieves better performance in most of different settings since that we consider more semantic information when masking out features.

Choice of block size The classification model: ResNet20 in our experiments has three groups of convolutional layers (res_1, res_2, res_3) after each of which the size of feature map is halved. Specifically, the size of feature map after each group are 32×32, 16×16, 8×8 respectively. In Fig. 4(b), we show the test accuracy of CIFAR-10 when applying CorrDrop after different groups of convolutional layers with different choices of block size (3, 5, 7, 9). We can gain the best performance when applying CorrDrop after the first group of convolutional layer with block size of 9. Since the size of feature map is decreased in deep convolutional layers, the receptive filed in deep layers is relatively larger than that in shallow layers, thus large block size in shallow layer and small block size in deep layers can better represent the local features. In our final implementation, we use block size of 9, 7, 5 after res_1, res_2, res_3 respectively.

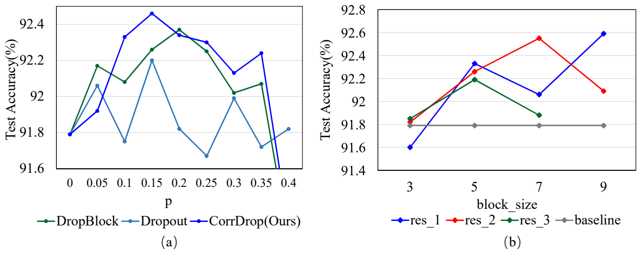


Fig. 4. Subfigure (a) is the test accuracy on CIFAR-10 with different choices of dropout probability. Subfigure (b) is the test accuracy on CIFAR-10 with different choices of block size.

4.5. Activation Visualization

We also utilize the class activation mapping(CAM) [27] to visualize the activation units of ResNet20 on images. From Fig. 5, DropBlock randomly drops some regions in the feature maps forcing the model to focus on a wide range of areas which generates a more distributed representation. However, CorrDrop regularizes the model by masking out those uncorrelated regions in the feature maps which encourages the model to focus on those meaningful regions for classification (e.g. the main object regions). In general, the activation map generated by model regularized with our method owes more compact representation and highly activated regions towards main object.

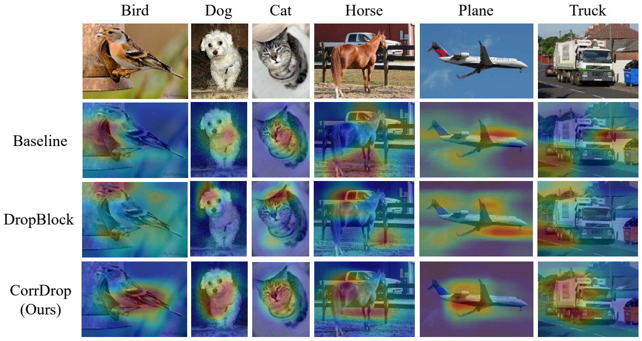


Fig. 5. Class activation mapping(CAM) [27] for ResNet20 trained with no regularization, DropBlock [12] and CorrDrop.

5. CONCLUSION

In this paper, we propose a novel and effective structural dropout variant CorrDrop to regularize CNNs, which remedies the problems of under- and over-dropping by considering the importance of discriminative regions. Extensive classification experiments demonstrate the superiority of CorrDrop compared with existing work and its applicability to different architectures. Moreover, the visualization of the activation map gives us an insight that our CorrDrop method can force the model to learn a compact and meaningful representation for classification.

6. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[6] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer, “Densenet: Implementing efficient convnet descriptor pyramids,” *arXiv preprint arXiv:1404.1869*, 2014.

[7] Anders Krogh and John A Hertz, “A simple weight decay can improve generalization,” in *Advances in neural information processing systems*, 1992, pp. 950–957.

[8] Lutz Prechelt, “Automatic early stopping using cross validation: quantifying the criteria,” *Neural Networks*, vol. 11, no. 4, pp. 761–767, 1998.

[9] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang, “Random erasing data augmentation,” *arXiv preprint arXiv:1708.04896*, 2017.

[10] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le, “Autoaugment: Learning augmentation policies from data,” *arXiv preprint arXiv:1805.09501*, 2018.

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le, “Dropblock: A regularization method for convolutional networks,” in *Advances in Neural Information Processing Systems*, 2018, pp. 10727–10737.

[13] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler, “Efficient object localization using convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656.

[14] Terrance DeVries and Graham W Taylor, “Improved regularization of convolutional neural networks with cutout,” *arXiv preprint arXiv:1708.04552*, 2017.

[15] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus, “Regularization of neural networks using dropconnect,” in *International conference on machine learning*, 2013, pp. 1058–1066.

[16] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger, “Deep networks with stochastic depth,” in *European conference on computer vision*. Springer, 2016, pp. 646–661.

[17] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le, “Learning transferable architectures for scalable image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[19] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang, “Residual attention network for image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.

[20] Sergey Zagoruyko and Nikos Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” *arXiv preprint arXiv:1612.03928*, 2016.

[21] Junsuk Choe and Hyunjung Shim, “Attention-based dropout layer for weakly supervised object localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2219–2228.

[22] Aaditya Prakash, James Storer, Dinei Florencio, and Cha Zhang, “Repr: Improved training of convolutional filters,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 10666–10675.

[23] Alex Krizhevsky, Geoffrey Hinton, et al., “Learning multiple layers of features from tiny images,” Tech. Rep., Citeseer, 2009.

[24] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, “On the importance of initialization and momentum in deep learning,” in *International conference on machine learning*, 2013, pp. 1139–1147.

[25] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[26] Sergey Zagoruyko and Nikos Komodakis, “Wide residual networks,” *arXiv preprint arXiv:1605.07146*, 2016.

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.