TARGETDROP: A TARGETED REGULARIZATION METHOD FOR CONVOLUTIONAL NEURAL NETWORKS

Hui Zhu[1,2], Xiaofang Zhao[1,3]

[1]Institute of Computing Technology, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, China
[3]Institute of Intelligent Computing Technology, Suzhou, Chinese Academy of Sciences, China

ABSTRACT

Dropout regularization has been widely used in deep learning but performs less effective for convolutional neural networks since the spatially correlated features allow dropped information to still flow through the networks. Some structured forms of dropout have been proposed to address this but are prone to result in over or under regularization as features are dropped randomly. In this paper, we propose a targeted regularization method, TargetDrop, which incorporates the attention mechanism to drop several discriminative feature units. Specifically, it masks out the target regions in the feature maps corresponding to the target channels. We conduct comprehensive experiments and demonstrate that TargetDrop outperforms the other dropout-based regularization methods.

***Index Terms*—** Dropout-based methods, Attention, Targeted regularization, Convolutional neural networks

1. INTRODUCTION

Convolutional neural networks (CNNs) have been widely used in computer vision and many excellent neural architectures [1, 2, 3] have been designed successively. In order to solve the performance degradation problems caused by over-fitting for larger-scale deep neural networks, many regularization methods have been proposed, such as weight decay, data augmentation [4, 5] and dropout [6].

However, the regularization effect of dropout for convolutional architectures is not as significant as that for fully connected networks because the spatially correlated features allow dropped information to still flow through CNNs [7]. To address this, some structured forms of dropout have been proposed to suppress the recovery of dropped information from the surrounding units, such as SpatialDropout [8], Cutout [4] and DropBlock [7]. But these methods are prone to result in over or under regularization as features are dropped randomly.

To make the regularization more intelligent, some researches [9, 10] combine structured dropout methods with attention mechanism. But they are prone to generate incomprehensible and separated masks or tend to lose the ability of capturing extensive features and make the model over-dependent on the most significant feature. Moreover, these methods only mask out the units in spatial dimension. They ignore the instructive information in channel dimension which is proven to be meaningful in CNNs [11], even in dropout-based regularization methods [7].

Fig. 1. Masks of naive Dropout [6], Dropblock [7] and our TargetDrop. The red regions denote the regions to be masked.

In this paper, we propose a novel regularization method named TargetDrop, which heuristically chooses target channels and then drops target regions in corresponding feature maps. As is shown in Fig.1, compared with naive Dropout and DropBlock which may lead to unexpected results by dropping randomly, TargetDrop is prone to precisely mask out several effective features of the main object, thus encouraging the model to learn more discriminative information. We conduct comprehensive experiments with commonly-used datasets CIFAR10, CIFAR100 and ImageNet on image classification and mini-ImageNet on few-shot learning tasks. The results and ablations demonstrate that TargetDrop can greatly improve the performance of CNNs by boosting the representation power of different architectures.

Our main contributions are summarized as follows:

• We propose a targeted regularization method, which incorporates attention mechanism to address the problem for unexpected results caused by dropping randomly.

• We propose the rule of choosing target channels and target regions, and further validate the regularization effect.

This work was supported by the National Key Research Program of China (Grant Nos. 2021YFF0703800).

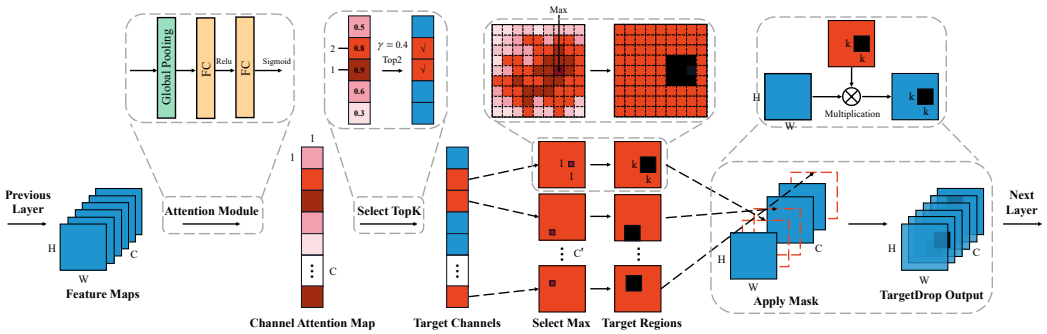

Fig. 2. The pipeline of TargetDrop. 1) Generate the channel attention map. 2) $topK$ elements are selected as the target channels according to γ. 3) Locate to a pixel with the maximum value in the feature map corresponding to each target channel and drop the $k \times k$ target region. 4) The mask is applied to the original feature maps by the multiplication operation.

2. RELATED WORK

Since Dropout [6] was proposed to improve the performace of networks by avoiding overfitting the training data, a series of drop-based regularization variations have emerged [12, 8, 13, 7, 9, 10, 14]. Besides, several attention processing methods are also related [11, 15, 16, 17]. As for few-shot learning, this work covers several classical methods [18, 19, 20].

3. METHODS

In this section, we propose our method TargetDrop, which mainly contains seeking out the target channels and regions. The pipeline of TargetDrop is shown in Fig. 1.

Target Channels. Given the output of the previous convolutional layer as $\boldsymbol{U} = [\boldsymbol{u_1}, \boldsymbol{u_2}, \cdots, \boldsymbol{u_C}] \in \mathbb{R}^{H \times W \times C}$, where H and W are the height and width of the feature map respectively, C is the number of channels. As each channel can be considered as a feature detector [21], random dropout may ignore 'what' is meaningful that channel attention focuses on and thus lead to uncertain results. Hence, we are eager to figure out the importance of each channel. We aggregate the spatial information of each feature map into channel-wise vector by using global average pooling which has been proven to work [11, 15]. This vector $\boldsymbol{v} \in \mathbb{R}^{1 \times 1 \times C}$ can be regarded as the statistic generated by shrinking through spatial dimensions $H \times W$ and this operation $\boldsymbol{F_{U \to v}}$ can be defined as:

$$v_c = \boldsymbol{F_{U \to v}}(\boldsymbol{u_c}) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \tag{1}$$

where v_c denotes the c-th element of $\boldsymbol{v}$. To further capture channel-wise dependencies, the vector is then forwarded to a shared network to produce the channel attention map $\boldsymbol{M} \in \mathbb{R}^{1 \times 1 \times C}$. The shared network is composed of two fully connected (FC) layers and two activation functions. Specifically, a dimensionality-reduction layer with parameters $\boldsymbol{W}_1 \in \mathbb{R}^{\frac{C}{r} \times C}$, a ReLU, a dimensionality-increasing layer with parameters $\boldsymbol{W}_2 \in \mathbb{R}^{C \times \frac{C}{r}}$ and then a Sigmoid function are connected alternately. Here, r is the reduction ratio to adjust the bottleneck. The map indicates the inter-channel relationships, and this operation $\boldsymbol{F_{v \to M}}$ can be defined as:

$$\boldsymbol{M} = \boldsymbol{F_{v \to M}}(\boldsymbol{v}, \boldsymbol{W}) = \sigma\left(\boldsymbol{W}_2 \delta\left(\boldsymbol{W}_1 \boldsymbol{v}\right)\right) \tag{2}$$

where δ and σ refer to the ReLU and Sigmoid, respectively. Then, we sort all the values in $\boldsymbol{M}$ and select the elements (tag "1" means to be selected) with top K values as the target according to the drop probability γ. Specifically, the channels corresponding to those elements marked as tag "1" in the vector $\boldsymbol{T} \in \mathbb{R}^{1 \times 1 \times C}$ are the target channels. Given the top K-th value in $\boldsymbol{M}$ as M_{topK} and this process can be described as:

$$K = \lfloor \gamma C \rfloor, \qquad T_p = \mathbb{1}\left(M_p \geq M_{topK}\right) \tag{3}$$

where M_p and T_p denote the p-th elements and $\mathbb{1}$ is an indicator function which equals to 1 iff the condition inside the bracket holds. $\lfloor \cdot \rfloor$ is the floor operation. Based on this, we further select target regions of original $H \times W$ feature maps corresponding to the target channels which we will elaborate in the following subsection.

Target Regions. After obtaining the target channels, we hope to further seek out a target region with much discriminative information that indicates 'where' is an informative part. Granted, utilizing spatial attention mechanism will work but is not necessary and may lead to considerable additional computation overhead. Considering the continuity of image pixel values [9], we design a low-cost yet effective strategy, which simply locate to a pixel with maximum value. Then, the other top values distributed in the surrounding continuous regions are considered as certain crucial features. Let h_1, h_2, w_1 and w_2 represent the boundaries of the target region, they are calculated based on the center location (a, b) and block size k:

$$\begin{aligned} h_1 &= a - \lfloor \tfrac{k}{2} \rfloor, h_2 = a + \lfloor \tfrac{k}{2} \rfloor \\ w_1 &= b - \lfloor \tfrac{k}{2} \rfloor, w_2 = b + \lfloor \tfrac{k}{2} \rfloor \end{aligned} \tag{4}$$

Then, we figure out the final mask exactly according to the target channels indicated in the vector $\boldsymbol{T}$ and the boundaries

of target regions. Specifically, the TargetDrop mask $\boldsymbol{S} = [\boldsymbol{s_1}, \boldsymbol{s_2}, \cdots, \boldsymbol{s_C}] \in \mathbb{R}^{H \times W \times C}$ can be described as:

$$s_q(m,n) = \begin{cases} 0 & T_q = 1 \wedge h_1 \leq m \leq h_2 \wedge w_1 \leq n \leq w_2 \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

where s_q and T_q denote the q-th elements of $\boldsymbol{s}$ and $\boldsymbol{T}$. Given the final output as $\widetilde{\boldsymbol{U}} = [\widetilde{\boldsymbol{u}}_{\boldsymbol{1}}, \widetilde{\boldsymbol{u}}_{\boldsymbol{2}}, \cdots, \widetilde{\boldsymbol{u}}_{\boldsymbol{C}}] \in \mathbb{R}^{H \times W \times C}$, we apply the mask and normalize the features:

$$\widetilde{\boldsymbol{u}}_{\boldsymbol{z}} = \boldsymbol{u}_z \odot \boldsymbol{s}_z \times \frac{numel(\boldsymbol{s}_z)}{sum(\boldsymbol{s}_z)} \tag{6}$$

where $\boldsymbol{u}_z$ and $\boldsymbol{s}_z$ denote the z-th elements, $numel(\cdot)$ counts the number of units, $sum(\cdot)$ counts the number of units with "1" and $\odot$ represents the point-wise multiplication operation.

TargetDrop. Combining the methods mentioned above, TargetDrop can select informative features and drop them intelligently, which is in sharp contrast to the aimless regularization methods. Without the dropped information, the model is encouraged to learn more discriminative features which may be not prominent before applying TargetDrop. Similar to other Dropout-based methods, TargetDrop is used to the outputs of the layers at the specific locations in CNNs and we do not apply TargetDrop during the inference phase.

4. EXPERIMENTS

In this section, we conduct comprehensive experiments on image classification and few-shot learning. Then, we report the performance and further show some analyses.

4.1. Image Classification

For image classification, we compare the regularization effect with other dropout-based methods and show the performance for different convolutional architectures.

Datasets. We test the commonly-used datasets CIFAR-10, CIFAR-100 [22] and ImageNet [23] for this part. Either CIFAR dataset contains 60,000 images of size 32×32 as well as ImageNet consists of 1,281,167 training images and 50,000 validation images. For preprocessing, we normalize the images and then apply a standard data augmentation scheme.

Training Setups. Networks using the official PyTorch implementation are trained on NVIDIA Tesla V100 GPUs. The batchsize is 128 for CIFAR and 1024 for ImageNet. The optimizer is SGD with Nesterov's momentum of 0.9 and the initial learning rate is set to 0.1. It will be decayed by the factor of 0.2 at 0.4, 0.6, 0.8 ratio of total epochs for CIFAR and of 0.1 at 0.3, 0.6, 0.9 ratio for ImageNet. For Cutout [4], the cutout size is 16×16 for CIFAR-10 and 8×8 for CIFAR-100. The reduction ratio r is 16, the drop probability and block size for TargetDrop are 0.15 and 5, respectively.

Comparison against other methods. For CIFAR, we use ResNet-18 and uniformly apply TargetDrop and baseline methods to the outputs of first two groups for a fair comparison. For ImageNet, following [7], we use ResNet-50 and apply TargetDrop to the outputs of last two groups. As is shown in Table 1, the results of our method outperform the competitors. In addition, we also test the orthogonality with Cutout [4] and the results are shown at the bottom two rows. Further improvements can be attributed to that TargetDrop is applied to the intermediate feature maps rather than the input.

Table 1. Comparison against the results (test errors (%)) of the state-of-the-art dropout-based regularization methods.

Methods	**ResNet-18**		**ResNet-50**
	CIFAR-10	**CIFAR-100**	**ImageNet** (Top1 / Top5)
No Regularization	4.72	22.46	23.7 / 6.9
Dropout [6]	5.14	23.82	-
DropBlock [7]	4.59	21.95	22.4 / 6.2
AutoAugment [5]	-	-	22.6 / 6.3
AttentionDrop [9]	4.51	21.53	-
TargetDrop (Ours)	**4.41**	**21.37**	**22.1 / 6.1**
Cutout [4]	3.99	21.96	-
Cutout + TargetDrop (Ours)	**3.67**	**21.25**	-

Table 2. The regularization effect (test errors (%)) of TargetDrop on CIFAR-10 with different convolutional architectures.

Networks	**Params**(M.)	Baseline	Dropout [6]	TargetDrop (Ours)
ResNet-20 [2]	0.27	8.21	7.80	**7.61**
VGG-16 [1]	14.73	6.17	6.43	**5.89**
WRN-28-10 [24]	36.48	4.02	4.04	**3.68**
Networks	DropPath [13]	Stochastic Depth [25]		DropBlock [7]
WRN-28-10 [24]	4.6	3.8		3.8

Regularization on different neural architectures. As is shown in Table 2, TargetDrop is an effective dropout-based regularization method and is applicable for the neural architectures of different scales. We supplement more results from [14] at the bottom two rows of the table with WRN-28-10 backbone, which is considered as a good match with CIFAR, to more fully reflect the effect of our method.

4.2. Few-Shot Learning

We further perform experiments on few-shot learning tasks. We follow a general practice to evaluate the effect with N-way K-shot and 15 query images. We utilize TargetDrop to improve different embedding networks, and then present the results on standard few-shot classification benchmarks.

Datasets. We use the most popular standard benchmark, mini-ImageNet [26], which is a derivative dataset of ILSVRC-2012 [23] for few-shot image classification. Mini-ImageNet consists of 100 randomly chosen classes and each class contains 600 images of size 84×84. We use the commonly-used split and adopt a standard data augmentation as in [20].

Training Setups. For embedding networks, our experiment covers 4 convolutional layers (Conv-4) and ResNet-12 backbones. Note that Conv-512F has used Dropout [6] and

Table 3. 5-way few-shot classification results (test accuracies (%)) with 95% confidence interval on mini-ImageNet with different backbones for 1-shot and 5-shot.

Methods	Backbones	5-way Acc.(%)	
		1-shot	5-shot
ProtoNets [18]	Conv-64F	49.42 ± 0.78	68.20 ± 0.66
ProtoNets + TargetDrop (Ours)	Conv-64F	**51.51 ± 0.65**	**69.29 ± 0.51**
R2D2 [19] w. Dropout	Conv-512F	51.2± 0.6	68.8 ± 0.1
R2D2 + TargetDrop (Ours)	Conv-512F	**54.18 ± 0.63**	**71.95 ± 0.49**
ProtoNets [18] w. DropBlock	ResNet-12	58.15 ± 0.70	75.49 ± 0.52
ProtoNets + TargetDrop (Ours)	ResNet-12	**58.34 ± 0.68**	**76.04 ± 0.51**

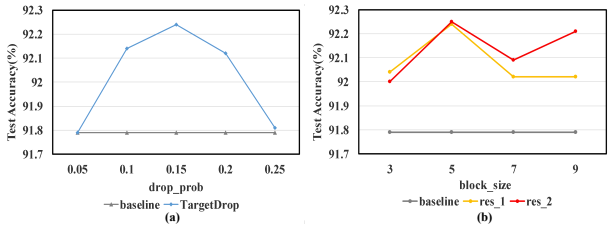

Fig. 3. (a) and (b) are the test accuracy on CIFAR-10 with different drop probabilities and block sizes, respectively.

ResNet-12 has used Dropblock [7]. For these backbones, we replace the original regularization methods with TargetDrop. Following [20], the optimizer is SGD with Nesterov momentum of 0.9 and weight decay of 5e-4. For each mini-batch, it consists of 8 episodes. All the models are meta-trained for 80 epochs with each epoch consisting of 1000 episodes. The learning rate is initially set to 0.1, and then adjusted to 0.003, 0.00032 and 0.00014 at epochs 12, 30 and 45, respectively.

Results. We utilize TargetDrop for three embedding backbones of different scales with two classical few-shot learning methods (*i.e.,* ProtoNets [18] and R2D2 [19]). Comparison against the baselines on 5-way mini-ImageNet is shown in Table 3. We can notice that the models achieve better performances compared with the baselines by leveraging our method. TargetDrop can yield better few-shot accuracy regardless of the feature dimension (Conv-64F is low (1600) and ResNet-12 is high (16000)). Thus, we argue that our method is robust and effective for few-shot learning by boosting the representation power of the embedding networks.

4.3. Analysis.

Parameters and Computation Overhead are limited. Additional parameters and computation mainly come from the channel attention mechanism. In addition, additional computation includes the simple selection of the maximum pixel. Specifically, the number of parameters in the training phase only increases by about 0.02% and the amount of computation increases similarly. While in the test phase, TargetDrop will be closed so that the complexity will not change.

Selection of Hyper-parameters. We further analyse the selection of the hyper-parameters mentioned above, which contain the drop probability γ and the block size k. We fix k to 5 and γ to 0.15, and apply TargetDrop to the output of first (two) group(s) to study another hyper-parameters. We perform the experiments on CIFAR-10 with ResNet-20 [2] and show the accuracies with varying hyper-parameters in Fig. 3. We can notice that our method improves the performance over the baseline in most settings. Besides, TargetDrop is more suitable for the intermediate feature maps with more channels and insensitive to the varying hyper-parameters.

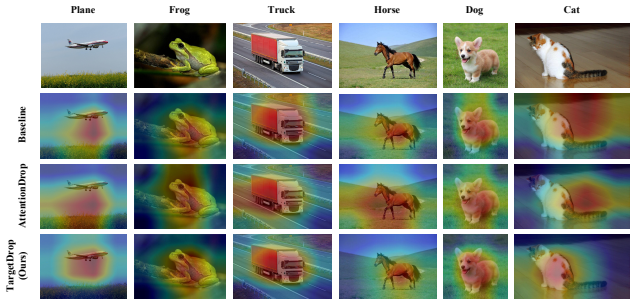

Fig. 4. Class activation mapping [27] for ResNet-18 trained with no regularization, AttentionDrop [9] and TargetDrop.

Activation Visualization. We utilize the class activation mapping (CAM) [27] to visualize the activation units of ResNet-18 [2] on several correctly classified images by each method. An intuitive grasp can be gained on how these regularizations encourage the model to capture and consider the features from the CAM visualization results, which are shown in Fig. 4. We can notice that the activation map generated by model regularized with TargetDrop shows strong competence in capturing extensive and relevant features towards the main object. Compared with others, the network regularized with TargetDrop tends to precisely focus on more discriminative information for image classification which we attribute to targeting and masking out certain effective features.

5. CONCLUSION

In this paper, we propose the novel regularization method TargetDrop for CNNs, which addresses the problem for unexpected results caused by the aimless methods by considering the importance of channels and regions within the feature maps. Our method is prone to precisely mask out several discriminative features of the main object, and thus encouraging the model to learn more effective information that may be less prominent before dropping. Extensive experiments and analyses demonstrate the outstanding performance of our method. In addition to the tasks covered in this work, we believe that TargetDrop is suitable for more datasets and tasks in the field of computer vision as long as CNNs are utilized.

6. REFERENCES

[1] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015*.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*.

[3] Mingxing Tan and Quoc V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*.

[4] Terrance Devries and Graham W. Taylor, "Improved regularization of convolutional neural networks with cutout," *CoRR*, 2017.

[5] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le, "Autoaugment: Learning augmentation strategies from data," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*.

[6] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, 2014.

[7] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le, "Dropblock: A regularization method for convolutional networks," in *Advances in Neural Information Processing Systems, NeurIPS 2018*.

[8] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler, "Efficient object localization using convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*.

[9] Zhihao Ouyang, Yan Feng, Zihao He, Tianbo Hao, Tao Dai, and Shu-Tao Xia, "Attentiondrop for convolutional neural networks," in *IEEE International Conference on Multimedia and Expo, ICME 2019*.

[10] Yuyuan Zeng, Tao Dai, and Shu-Tao Xia, "Corrdrop: Correlation based dropout for convolutional neural networks," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020*.

[11] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*.

[12] Li Wan, Matthew D. Zeiler, Sixin Zhang, Yann LeCun, and Rob Fergus, "Regularization of neural networks using dropconnect," in *the 30th International Conference on Machine Learning, ICML 2013*.

[13] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le, "Learning transferable architectures for scalable image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*.

[14] Hieu Pham and Quoc V. Le, "Autodropout: Learning dropout patterns to regularize deep networks," in *35th AAAI Conference on Artificial Intelligence, AAAI 2021*.

[15] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "CBAM: convolutional block attention module," in *15th European Conference on Computer Vision (ECCV 2018)*.

[16] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang, "Selective kernel networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*.

[17] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon, "BAM: bottleneck attention module," in *British Machine Vision Conference 2018, BMVC 2018*.

[18] Jake Snell, Kevin Swersky, and Richard S. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems, NeurIPS 2017*.

[19] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi, "Meta-learning with differentiable closed-form solvers," in *International Conference on Learning Representations (ICLR)*, 2019.

[20] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto, "Meta-learning with differentiable convex optimization," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*.

[21] Matthew D. Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *13th European Conference on Computer Vision (ECCV 2014)*.

[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Tech. Rep*, 2009.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, 2015.

[24] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference, BMVC 2016*.

[25] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger, "Deep networks with stochastic depth," in *14th European Conference on Computer Vision (ECCV 2016)*.

[26] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems, NeurIPS 2016*.

[27] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.