applied sciences

MDPI

Article

FocusedDropout for Convolutional Neural Network

Minghui Liu [1], Tianshu Xie [1], Xuan Cheng [1], Jiali Deng [1], Meiyi Yang [2], Xiaomin Wang [2,*] and Ming Liu [1]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; minghuiliu@std.uestc.edu.cn (M.L.); tianshuxie@std.uestc.edu.cn (T.X.); cs_xuancheng@std.uestc.edu.cn (X.C.); dengjiali@std.uestc.edu.cn (J.D.); csmliu@uestc.edu.cn (M.L.)

[2] Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou 324003, China; meiyiyang@std.uestc.edu.cn

* Correspondence: xmwang@uestc.edu.cn

Featured Application: We propose a non-random dropout method named FocusedDropout, aiming to make the network focus more on the target. It can effectively improve the performance of feature learning in deep learning that can be used for any applications with deep learning technology.

Abstract: In a convolutional neural network (CNN), dropout cannot work well because dropped information is not entirely obscured in convolutional layers where features are correlated spatially. Except for randomly discarding regions or channels, many approaches try to overcome this defect by dropping influential units. In this paper, we propose a non-random dropout method named FocusedDropout, aiming to make the network focus more on the target. In FocusedDropout, we use a simple but effective method to search for the target-related features, retain these features and discard others, which is contrary to the existing methods. We find that this novel method can improve network performance by making the network more target focused. Additionally, increasing the weight decay while using FocusedDropout can avoid overfitting and increase accuracy. Experimental results show that with a slight cost, 10% of batches employing FocusedDropout, can produce a nice performance boost over the baselines on multiple datasets of classification, including CIFAR10, CIFAR100 and Tiny ImageNet, and has a good versatility for different CNN models.

Keywords: classification; convolutional neural network; dropout; regularization

Citation: Liu, M.; Xie, T.; Cheng, X.; Deng, J.; Yang, M.; Wang, X.; Liu, M. FocusedDropout for Convolutional Neural Network. *Appl. Sci.* **2022**, *12*, 7682. https://doi.org/10.3390/app12157682

Academic Editor: Krzysztof Koszela

Received: 23 June 2022
Accepted: 27 July 2022
Published: 30 July 2022

1. Introduction

In recent years, deep neural networks have made significant achievements in many computer vision tasks such as image classification [1–4], object detection [5–7], and semantic segmentation [8,9]. However, deep layers and millions of neurons also lead to inadequate training of CNN. Dropout [10] is proposed as a regularization method widely used to fight against overfitting, which stochastically sets the activations of hidden units to zero during training. For deep CNN, dropout works well in fully connected layers, but its effect is still not apparent in convolutional layers, where features are correlated spatially. When the features are strongly correlated between adjacent neurons, the information of discarded neurons cannot be completely obscured.

Many researchers have observed this defect and tried to make dropout better regularize CNN. As shown in Figure 1, SpatialDropout [11] randomly discards entire channels from whole feature maps. DropBlock [12] randomly discards units in a contiguous region of a channel instead of substantive units. Guided dropout [13], AttentionDrop [14], and CamDrop [15] search the influential units in the network through different methods and drop them to enhance the generalization performance of the network. Furthermore, Auto Dropout [16] is proposed to learn the dropping patterns of SpatialDropout and DropBlock via reinforcement learning. Although it achieves state-of-the-art results, it requires a huge computational cost and is more like an extension of the mentioned approaches.

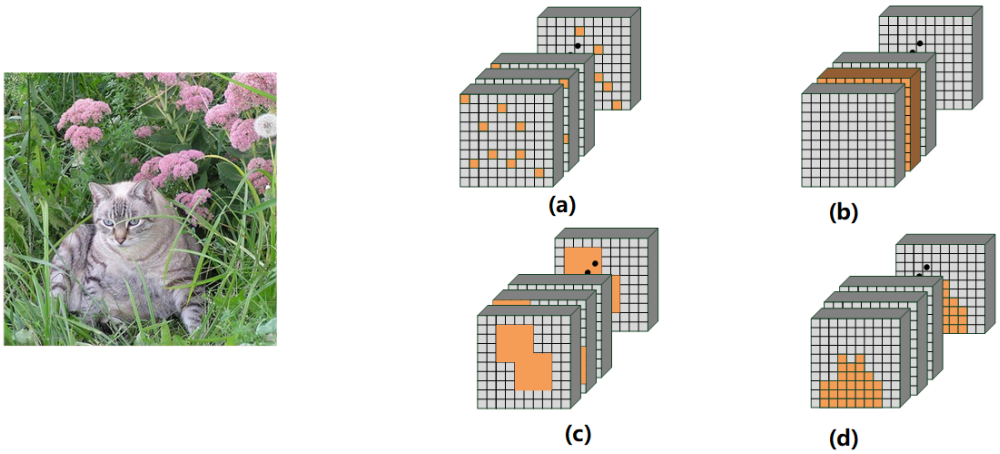


Figure 1. Four forms of dropout in CNN. (**a**) Dropout randomly drops units on each channel. However, it does not work well on CNN. (**b**) SpatialDropout randomly drops channel instead of units. (**c**) DropBlock randomly drops some contiguous regions inspired by Cutout. (**d**) Some non-random dropout methods discard influential units. The orange neurons, channels and contiguous regions present the discarded parts, and the gray will be retained.

From another perspective, the foreground of the image is the main focus of classification, which means accurately finding the foreground is important for image classification. The results of observation experiments show the spatial information of images remains unchanged during training, and channels in the network are associated with the classification target. This indicates that we can enhance the foreground and thus improve the recognition of objects. It can also avoid noise interference from backgrounds. All of them enlightened us on using dropout to enhance the network's ability to recognize the objects by keeping the foreground-related units and dropping other units during training, which is contrary to the existing methods that drop influential ones. There are two challenges: how to identify the foreground-related units, and how to prevent overfitting when enhancing influential ones.

In this paper, we propose FocusedDropout as a non-random dropout method to regularize CNNs. Inspired by Network Dissection [17], a visualization method proposed by Hence, Bau et al., we first select the channel with the highest average activation value and discard the units with an activation value below a threshold on this channel. Then, we can distinguish useful units from redundant units with the support of spatial invariance of CNN by discarding all units that have the same positions as previously discarded ones on the rest of the channels. Increasing weight decay [18] while using FocusedDropout can avoid overfitting and increase accuracy. As a result, the network focuses more on the units that have the highest probability associated with the target, which is particularly effective for regularization. Extensive experimental results show that even a slight cost, 10% of batches employing FocusedDropout, can produce a nice performance boost over the baselines on CIFAR10 [19], CIFAR100 [19], Tiny ImageNet (http://tiny-imagenet.herokuapp.com/), and has a good versatility for different CNN models including ResNet [4], DenseNet [20], VGGNet [21] and Wide ResNet [22].

This work provides these primary contributions: 1. This paper proposes a non-random dropout method to regularize CNNs, called FocusedDropout. Unlike previous works, which discarded influential units, it enhances features related to the classification by discarding others, thus effectively improving the classification performance. 2. We also propose an assisted training strategy to avoid overfitting, magnifying the weight decay while the FocusedDropout is used, and only randomly select 10% of the batches to use the proposed method. 3. Extensive experiments are conducted to verify the performance. The results show the FocusedDropout is lightweight, achieving the best score in many tasks.

2. Related Work

Regularization methods have always been effective ways to improve the performance of neural networks, which involve two common regularization techniques: data augmentation and dropout [10]. Unlike data augmentation, such as [23–25], which expands the data space, dropout injects noise into feature space by randomly zeroing the activation function to avoid overfitting. Inspired by dropout, DropConnect [26] drops the weights of the networks instead of activations to regularize large neural network models. Shakeout [27] randomly chooses to enhance or inverse the contributions of each unit to the next layer. Concrete Dropout [28] utilizes concrete distributions to generate the dropout masks. Alpha-dropout [29] is designed for Scaled Exponential Linear Unit activation function. Variational dropout [30] is proposed as a generalization of Gaussian dropout, but with a more flexibly parameterized posterior. These strategies work well on the fully connected network, but their performance on CNN is unsatisfactory. The reason is that the correlation between the units on each channel in CNN is so strong that information can still flow through the networks despite dropout. Obviously, this reduces the improvement [11].

Previous studies demonstrated that the correlation always relies on location information, which pushes forward the study about dropping the network weights by leveraging structures. SpatialDropout [11] is proposed by randomly discarding the entire channels rather than individual activations. Channel-Drop [31] overcomes the drawbacks of local winner-takes-all methods used in deep convolutional networks. Inspired by Cutout [32], DropBlock [12] randomly drops some contiguous regions of the feature map. With the continuous improvement of the network models, lots of regularization methods, such as Stochastic Depth [33], Shake-Shake [34], and ShakeDrop [35], have emerged for specific CNNs such as ResNet [4] and ResNext [36]. Although the above methods enhance the networks' ability by randomly discarding neurons or changing weight, the network also easily suffers from the overfitting problem because the networks can still learn too many features about the objects during the training.

Therefore, some approaches focus on discarding specific units are proposed. Maxdrop [37] drops the activations which have high values across the feature map or the channels. Targeted dropout [38] combines the concept of network cropping with dropout, which selects the least important weights in each round of training and discards the candidate weights to enhance network robustness. Guided Dropout [13] defines the strength of each node and strengthens the training of weak nodes by discarding strong nodes during training. Weighted Channel Dropout [39] aims to solve the overfitting in small data set training, which calculates the average activation value of each channel and retains the high-valued channels with higher possibility. AttentionDrop [14] drops features adaptively based on attention information. CamDrop [15] electively abandons some specific spatial regions in predominating visual patterns by considering the intensity of class activation mapping. Discarding influential units makes the network less prone to the overfitting problem. However, it cannot improve the classification performance of the network.

Recently, some algorithms have also been proposed, and had good results. R-Drop [40] offers a simple consistency training strategy to regularize dropout. It forces the output distributions of different sub-models generated by dropout to be consistent with each other. Contextual Dropout [41] proposes a simple and scalable sample-dependent dropout, which learns the dropout probabilities with a variational objective and is compatible with both Bernoulli and Gaussian dropout. Juan Shu et al. [42] develop a Heteroscedastic Gaussian Dropout algorithm, where the dropout probability is determined by another model with mirrored GNN architecture. On the other hand, with the development of reinforcement learning, it has been applied to many domains, including dropout. A controller in Auto Dropout [16] is introduced to learn the dropping patterns of SpatialDropout and DropBlock, and generate them at every channel and layer, which will be followed in the training procedure of the target network. Significantly, although those methods achieve improvement

with varying degrees, our method is able to match or outperform their scores. Additionally, the computational cost increment is lower than others.

3. Motivation

Existing non-random dropout methods judge units' influence in different ways and discard the units with greater influence, aiming to make the network more robust from the perspective of preventing overfitting. However, influential units contain information conducive to classification, and strengthening the learning of this part of units may make networks focus more on target information. The network could learn more features about the target point to the classification, thus improving the performance benefits of more precise learning. However, this may also lead to overfitting performance because the network will learn many specific characteristics rather than the general ones that indicate the semantics of said domain. Therefore, there are two challenges: one is how to identify the target-related units, and the other is how to prevent overfitting when enhancing influential ones.

We select the target-related units based on Network Dissection [17], a visualization method for the interpretability of latent representations of CNN. In this study, after putting an image containing pixel-wise annotations for the semantic concepts into a trained network, the activation map of the target neuron is scaled up to the resolution of the ground truth segmentation mask. If the measurement result of alignment between the upsampled activation map and the ground truth segmentation mask is larger than a threshold, the neuron would be viewed as a visual detector for a specific semantic concept. The results show the visual detectors always correspond to the ground truth of the target. From this work, we can conclude that the image's spatial information remains unchanged during training, and there are channels related to the classification target in the trained network. Next, we need to find the target-related channels.

We conduct an exploratory experiment to find the target-related units. We first put the CIFAR10 dataset's validation set containing 10,000 images into the vanilla ResNet-56 model pretrained with CIFAR10, and record the channel with the highest activation value in the last layer for every successfully classified image. Note that there are 9392 successfully classified images in 10 categories, and the number of the channel in the last layer is 256. The result can be seen in Figure 2. We find that the channels with the highest activation value are different for different categories. For the same category of successfully classified images, there are only one or two fixed channels with the highest activation values. This shows that the channel with the highest activation value is crucial to the success of the classification task. Thus, we speculate that the channel with the highest average activation value has the highest probability of containing foreground information, and the units with the high activation value on this channel have the highest probability of representing foreground features. Therefore, we can take this channel as a reference, and combine spatial invariance to select other useful units on the remaining channels.

As mentioned, retaining only the units with larger weights may exacerbate overfitting intuitively. Thus, we want to find a balance between strengthening the network's attention to the target and preventing the network from overfitting. To achieve this balance, we propose two countermeasures. First, we only randomly select 10% of the batches to use FocusedDropout during every training epoch. Additionally, we will magnify weight decay when using FocusedDropout. We find these two measures can avoid overfitting effectively and improve the robustness of the network.

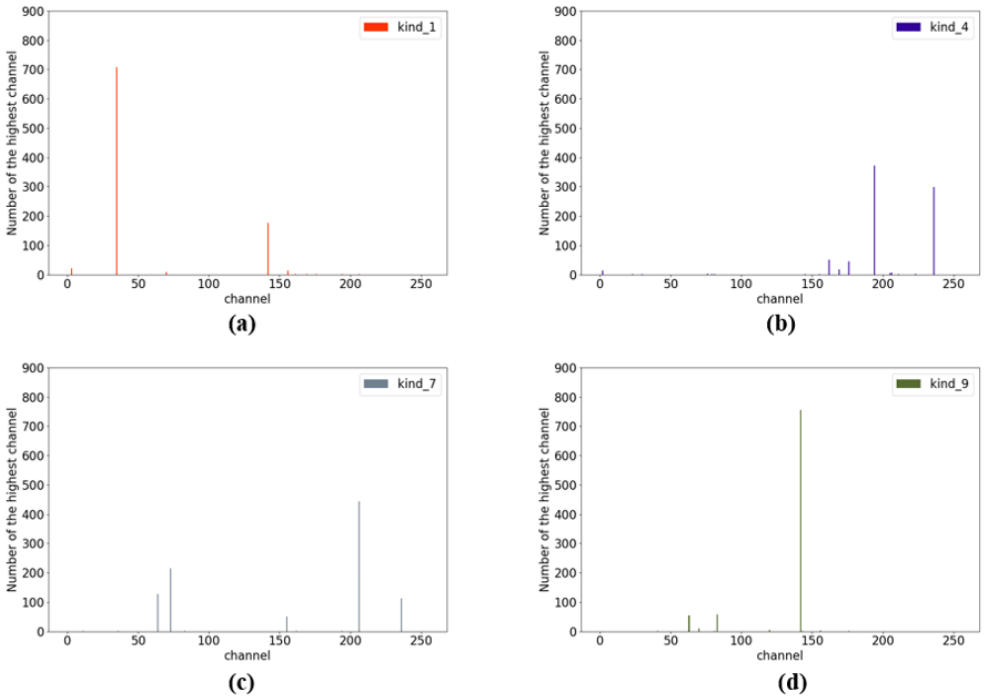


Figure 2. (**a**–**d**) The number of channels with the highest average activation value for 10,000 images in the trained ResNet-56 model's last layer for the images of four categories. We randomly choose four categories in CIFAR10. The x-axis shows the specific channel, and the y-axis measures the number of images in the corresponding channel with the highest activation value. The channels with the highest activation value are different for different categories of images, but there are one or two fixed channels with the highest activation values for the successfully classified images in the same category.

4. Our Approach

FocusedDropout is a highly targeted approach that makes the networks focus on the foreground rather than the background or noise. The main idea of FocusedDropout is to keep the units in the preferred locations related to the classification target and drop the units in other locations. According to the spatial invariance of CNN, the locations of images' features are fixed during training so that different channels' units with the same spatial positions represent the same image features. Inspired by this phenomenon, FocusedDropout uses a binary mask to cover the target-independent units with the same positions on each channel, as demonstrated in Figure 3. The algorithm of FocusedDropout is illustrated in Algorithm 1. Next, we will present the details of FocusedDropout.

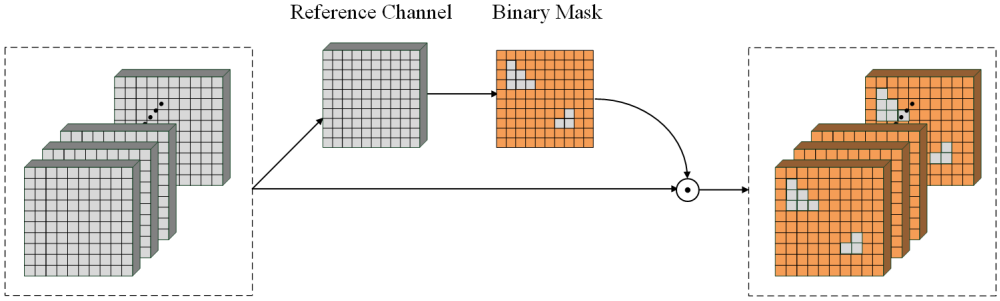


Figure 3. Illustration of FocusedDropout. The channel with the highest activation value is selected as the reference channel to generate the binary mask. After being covered by the mask, the units most likely to be related to the target will be retained, and the others will be discarded. The orange represents the neurons dropped by FocusedDropout.

Algorithm 1 FocusedDropout

Require: whole channels of the previous layer $C = [c_1, c_2, c_3, \ldots, c_n]$, mode

if mode = Inference, **then**
  Return $C^* = C$
end if
Calculate the average activation value for each c_i as w_i, Get the c_k with the max w_i
Get the unit having the highest activation value as $c_k(\overline{x}, \overline{y})$
Make the all-zero mask m having the same size with c_i
for $m(i, j)$ in m **do**
  if $c_k(i, j) > random(0.6, 0.9) \cdot c_k(\overline{x}, \overline{y})$ **then**
    $m(i, j) = 1$
  end if
end for
Return $C^* = C \odot m$

4.1. Selecting the Reference Channel

Each channel output by convolutional layers can be regarded as a set of features extracted from the image. Our goal is to find the channel with the highest possibility of obtaining the features relevant to the target. We observe that the channel with the largest average activation value has the greatest effect on the result, and we consider that it contains the most important features. Therefore, FocusedDropout uses Global Average Pooling to acquire the average activation value of each channel. To facilitate the presentation, we introduce the following concepts. $C = [c_1, c_2, \ldots, c_n]$ denotes the whole channels from the previous layer; c_i denotes the single channel, i.e., the entire feature map; and $c_i(x, y)$ denotes the activation value of a unit on c_i. So, the average activation value of c_i is computed as

$$w_i = \frac{1}{w \times h} \sum_{x=1}^{w} \sum_{y=1}^{h} c_i(x, y), \tag{1}$$

where w and h are the shared width and height of all channels. After getting all the weights, we choose the max one, c_k, as the reference channel, and k is computed as

$$k = \underset{i}{argmax}\, w_i \tag{2}$$

4.2. Finding the Focused Area

After obtaining the reference channel, we need to get the location information of the target-related features from it. The units with high activation values have more influence on the task, so first, we find the highest activation value $c_k(\overline{x}, \overline{y})$ on the reference channel as the criterion for setting the threshold, where $(\overline{x}, \overline{y})$ is set as

$$(\overline{x}, \overline{y}) = \underset{(x,y)}{argmax}\, c_k(x, y) \tag{3}$$

We take a random ratio γ between 0.3 and 0.6 and set the threshold as

$$threshold = \gamma \cdot c^*(\overline{x}, \overline{y}) \tag{4}$$

We set a random ratio rather than a certain one, because different images have different numbers of features. Units with an activation value higher than the threshold are considered target-related. The assembling of these units' positions is considered as the area related to the target, i.e., the focused area A.

4.3. Generating the Binary Mask

Although the focused area is obtained from the reference channel, units in the same area on the other channels are still related to the target due to the spatial invariance of CNN. In other words, different channels from the previous layer extract various features from the image, but the same position of these channels represents the same location of the image. Thus, we need to retain each channel's units in the preferred area. To simplify the calculation, we use a binary mask multiplying all channels to achieve this step. The binary mask m satisfies

$$m(i,j) = \begin{cases} 0, (i,j) \notin A \\ 1, (i,j) \in A \end{cases} \tag{5}$$

where $m(i,j)$ represents the value of the binary mask, and A represents the focused area. Only the units contained in the focused area can be reserved when the binary mask is multiplied by all channels.

4.4. Magnifying Weight Decay (MWD)

During training, we randomly choose 10% of batches using FocusedDropout rather than every batch. Only retaining the high-weight units may aggravate the overfitting, so we increase weight decay when using FocusedDropout, which can limit the network's over attention to the classification target and inhibit the overfitting. This is a new training trick for dropout, because dropout is often used in all batches of every epoch and weigh decay is fixed during training. We call the rate of batches applying FocusedDropout *participation_rate*. The computing resource and time consumption can be decreased obviously, and the model's performance can still be enhanced with this trick. In the testing phase, we retain all units as conventional dropout.

4.5. Summing Up

In summary, we first find the focused areas by searching the reference channel with the highest activation value. They will be considered units, with the higher activation value associated with the mentioned reference channel. Then, we drop backgrounds by generating a binary mask and multiplying it with all channels. In this way, the network will focus on the objects, thus improving the performance. However, it may also lead to the network falling into the overfitting trap. Therefore, the weight decay will be magnified while the FocusedDropout is used, and only 10% of the batches are selected to use the proposed method.

5. Experimental Section

In the following sections, we investigate the effectiveness of FocusedDropout, mainly for different computer vision tasks. We apply FocusedDropout to different networks for image classification on various kinds of datasets. Additionally, we study the effect of our method on object detection.

5.1. Implementation Details

To evaluate our method's generalization, we apply FocusedDropout to three classification datasets: CIFAR10, CIFAR100, and Tiny ImageNet. Standard data augmentation schemes such as flipping and random cropping are also incorporated. The hyperparameters, including participation_rate and γ, are set by cross-validation. For Wide ResNet-28, the learning rate is decayed by the factor of 0.1 at 60,120,160; for other networks, the learning rate is decayed by the factor of 0.1 at 150,225. DropBlock [12] is applied to the output of the first two groups. Dropout [10], SpatialDropout [11], and FocusedDropout are applied to the output of the penultimate group for fairness of comparison. The highest validation accuracy over the full training course is chosen as the result. All experiments are performed with Pytorch [43] on Tesla M40 GPUs.

5.2. Evaluation of FocusedDropout

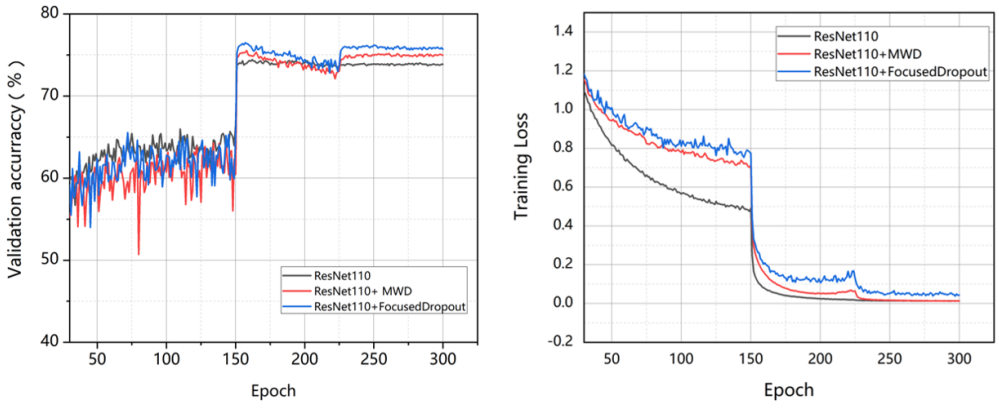
5.2.1. CIFAR10 and CIFAR100

The CIFAR10 [19] dataset consists of 60,000 32×32 color images of 10 classes, each with 6000 images, including 50,00 training images and 10,00 test images. We adopt ResNet-20, ResNet-56, ResNet-110 [4], VGGNet-19 [21], DenseNet-100 [20], and Wide ResNet-28 [22] as the baselines to evaluate FocusedDropout's generalization for different structures of networks. We set γ to a random value between 0.3 and 0.6, *participation_rate* = 0.1. As shown in Table 1, FocusedDropout achieves better improvement than other regularization methods in all networks. Generalization is an essential property of regularization methods. Experiments show that our method is suitable for networks with different layers, channels, and parameter spaces.

Table 1. Test accuracy (%) on CIFAR10 dataset using CNN architectures of ResNet-20, ResNet-50, ResNet-110,VggNet-19, Densenet-100,Wide ResNet-28 (Top accuracy is in bold). We report the average over 3 runs.

Method	ResNet-20	ResNet-56	ResNet-110	VggNet-19	DenseNet-100	WRN-28
Baseline	91.48 ± 0.21	93.84 ± 0.23	94.32 ± 0.16	93.55 ± 0.24	95.30 ± 0.12	96.16 ± 0.06
Dropout	91.51 ± 0.31	93.74 ± 0.11	94.52 ± 0.23	93.14 ± 0.30	95.32 ± 0.08	96.07 ± 0.04
SpatialDropout	91.98 ± 0.13	94.28 ± 0.20	94.76 ± 0.15	93.19 ± 0.12	95.40 ± 0.16	96.13 ± 0.02
DropBlock	92.15 ± 0.14	94.01 ± 0.08	94.92 ± 0.07	93.50 ± 0.04	95.40 ± 0.31	96.21 ± 0.03
FocusedDropout	**92.08 ± 0.17**	**94.67 ± 0.07**	**95.24 ± 0.15**	**93.74 ± 0.19**	**95.57 ± 0.16**	**96.48 ± 0.13**

The CIFAR100 [19] dataset has the same number of images but 100 classes, which means the training of CIFAR100 is harder than CIFAR10 because it has more image types and less training data for each kind. Results are summarized in Table 2. FocusedDropout still performs better than other dropout methods, and for ResNet-110, the promotion can even exceed 1.5%. Validation accuracy and training loss comparison of ResNet-110 on CIFAR100 can be seen in Figure 4, which shows that random MWD and FocusedDropout can effectively prevent overfitting.



(a) Validation accuracy comparison **(b) Training loss comparison**

Figure 4. Comparison of the validation accuracy curves and training loss curves of ResNet-110 on CIFAR100. The performance of the training networks with FocusedDropout and random Magnify Weight Decay (MWD) are worse than the baseline before epoch 150, but when the learning rate decays, the networks' performance is much better than the original model.

Table 2. Test accuracy (%) on CIFAR100 dataset. All settings are the same as on CIFAR10. We report the average over 3 runs.

Method	ResNet-20	ResNet-56	ResNet-110	VggNet-19	DenseNet-100	WRN-28
Baseline	69.85 ± 0.20	73.71 ± 0.28	74.71 ± 0.23	73.14 ± 0.16	77.25 ± 0.20	81.27 ± 0.20

Table 2. *Cont.*

Method	ResNet-20	ResNet-56	ResNet-110	VggNet-19	DenseNet-100	WRN-28
Dropout	69.81 ± 0.12	73.81 ± 0.27	74.69 ± 0.33	73.01 ± 0.21	77.45 ± 0.12	81.21 ± 0.11
SpatialDropout	69.71 ± 0.05	74.38 ± 0.17	74.76 ± 0.12	72.84 ± 0.21	77.97 ± 0.16	81.40 ± 0.30
DropBlock	70.03 ± 0.11	73.92 ± 0.10	74.92 ± 0.07	73.01 ± 0.04	77.33 ± 0.09	81.15 ± 0.09
FocusedDropout	**70.27 ± 0.06**	**74.97 ± 0.16**	**76.43 ± 0.17**	**73.91 ±0.19**	**78.35 ± 0.16**	**81.90 ± 0.11**

5.2.2. Tiny ImageNet

Tiny ImageNet dataset is a subset of the ImageNet [44] dataset with 200 classes. Each class has 500 training images, 50 validation images, and 50 test images. All images are of 64 × 64 resolution. The test-set label is not publicly available, so we use the validation set as a test set for all the experiments on Tiny ImageNet, following the common practice. We apply different dropout methods to ResNet-110 and compare the results with FocusedDropout. As shown in Table 3, FocusedDropout achieves a significant 2.16% improvement over the base model and has a larger gain compared to other dropout methods.

Table 3. The performance comparison on Tiny ImageNet dataset. The best accuracy is achieved by FocusedDropout. We report the average over 3 runs.

Metod	Validation Accuracy (%)
ResNet-110	62.42 ± 0.25
ResNet-110 + Dropout	62.32 ± 0.05
ResNet-110 + SpatialDropout	62.55 ± 0.10
ResNet-110 + DropBlock	63.13 ± 0.29
ResNet-110 + FocusedDropout	**64.58 ± 0.16**

To get an intuitive grasp of how FocusedDropout makes the network focus more on target, we use class activation mapping (CAM) [45] to visualize the activation units of ResNet-110 trained by Tiny ImageNet, as shown in Figure 5, from which we observe that the network trained with FocusedDropout can more accurately identify the target's location and contour; additionally, the identification of the main target and the secondary target in the image is clearer. This proves that our approach enables the network to focus more on the target. It also shows that making the network more goal oriented helps improve the performance of the network.

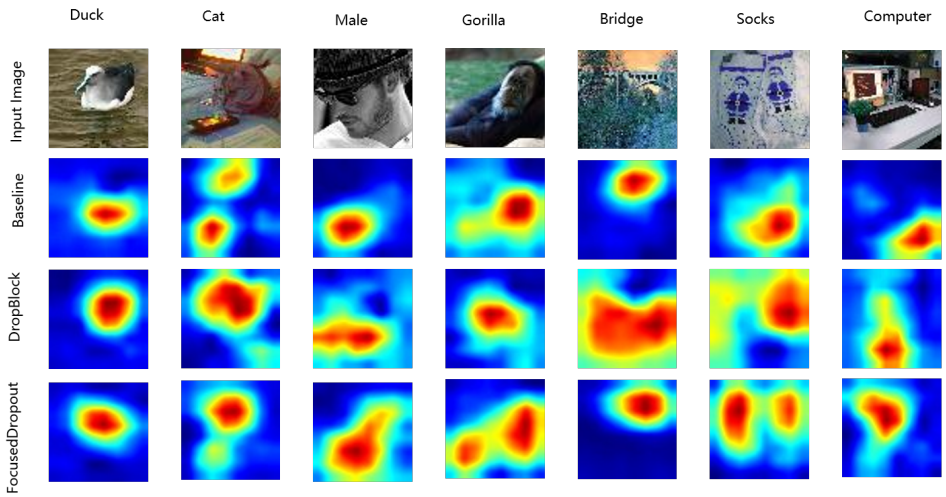


Figure 5. Class activation mapping (CAM) [45] for ResNet-110 model trained without any method, trained with DropBlock, and trained with FocusedDropout. The model trained with our method can accurately identify the target's location and contour.

5.3. Object Detection in PASCAL VOC

In this subsection, we show FocusedDropout can also be applied to the training object detector in Pascal VOC [46] dataset. RetinaNet [47] is used as the framework composed of a backbone network and two task-specific subnetworks for the experiments. The ResNet-50 backbone, which is responsible for computing a convolutional feature map over an entire input image, is initialized with the ImageNet-pretrained model and then fine-tuned on Pascal VOC 2007 and 2012 trainval data. We apply FocusedDropout to ResNet-50 in ResNet-FPN. The accuracy of ResNet-50 trained with FocusedDropout on ImageNet achieves 76.84%, 0.52% higher than the baseline (76.32%). Models are evaluated on VOC 2007 test data using the mAP metric. We follow the fine-tuning strategy of the original method.

As shown in Table 4, the model pre-trained with FocusedDropout achieves 71.05%, +0.91% higher than the baseline performance. It shows that the model trained with FocusedDropout can identify the target more easily and capture the position of the object more accurately.

Table 4. The performance comparison on Pascal VOC dataset.

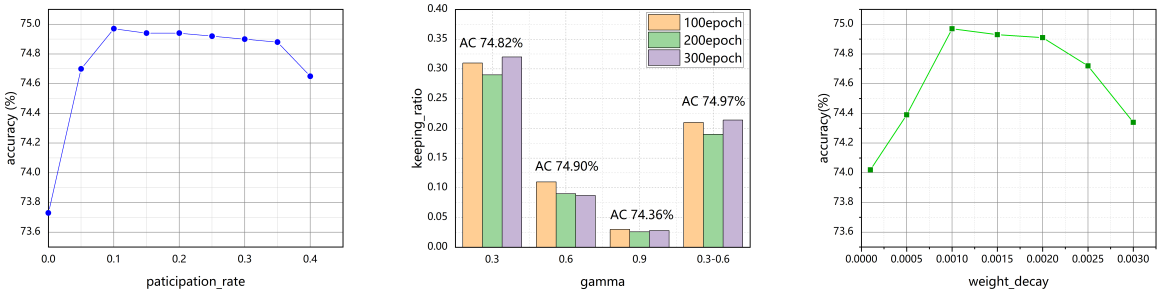
Method	mAP (%)
RetinaNet	70.14 ± 0.17
RetinaNet + FocusedDropout pretrained	**71.05** ± **0.13**

5.4. Ablation Study

In this subsection, we take extensive experiments to analyze the behaviors of FocusedDropout.

5.4.1. Effect of the Participation _RATE

We explore the effect of the *participation_rate* on classification accuracy. Taking the performance of ResNet-56 on CIFAR100 as an example, we change the setting of *participation_rate* from 0 to 0.4. As shown in Figure 6a, with the increase in *participation_rate*, the accuracy first rises and then decreases, after reaching the highest point when the *participation_rate* is 0.1. This indicates that increasing the *participation_rate* moderately can improve the network's performance, but excessive use of FousedDropout may lead to the deviation of the information learned by the network. The change is not apparent when *participation_rate* is below 0.3, so the parameter setting is not crucial for our method. The main additional computational costs come from the generation of the mask, which only requires some simple matrix operations. Increasing the *participation_rate* will add the cost, but is still closed to other methods. The additional computational costs can almost be ignored in the standard setting.



(**a**) (**b**) (**c**)

Figure 6. CIFAR100 validation accuracy against particitpation_rate, γ and weight decay with ResNet-56 model. (**a**) participation_rate; (**b**) γ; (**c**) weight decay.

5.4.2. Analysis of Keeping _RATIO

An obvious characteristic of FocusedDropout is that we do not directly set the *keeping_ratio*, but choose a threshold to filter the units. We perform some experiments to explore the *keeping_ratio* with a different threshold. Taking the performance of ResNet-56 on CIFAR100 as an example, we set γ to 0.3, 0.6, 0.9, or a random number between 0.3 and 0.6. The ratio of the number of zeros on the binary mask to the total number of the mask's units is considered the *keeping_ratio*. The mask generated by each image is different, so we take the average of every *keeping_ratio* that appeared in an epoch as the *keeping_ratio* of this epoch. As illustrated in Figure 6b, the number of units discarded by FocusedDropout greatly exceeds the conventional dropout when γ is greater than 0.3. Additionally, Figure 6b shows how the threshold γ affects accuracy with different values or ranges. When γ is set to a random number, the performance is better than a fixed value. We consider different images have different features, so a random number is better than a fixed value.

5.4.3. Exploration of Weight Decay

We tested the effect of magnifying weight decay (MWD) when using FocusedDropout. As shown in Table 4, the performance of the network with higher weight decay is better. We consider that higher weight decay can limit the changing parameters, and the network will not focus too much on the target. Another notable phenomenon is with the same probability, only randomly expanding weight decay also improves the network's performance, but when combined with FocusedDropout, the performance improvement is even greater than the sum of the two. This demonstrates that FocusedDropout is a balance between making the network pay more attention to the target and preventing overfitting. Figure 6c shows that with the increase in weight decay, the accuracy first rises and then decreases after reaching the highest point when the weight decay is 0.001.

5.4.4. Comparison with Opposite Approach

We explored the performance of methods using the opposite approach to discard units. When training ResNet-56 on CIFAR100, we set the units included in the focused area on the binary mask to one, and other positions to zero. Meanwhile, we keep the same parameter settings as FocusedDropout for training. The comparison can be seen in Table 5.

Table 5. Comparisons among baseline, randomly magnifying weight decay, FocusedDropout without magnifying weight decay, opposite approach and FocusedDropout of CIFAR100 on ResNet-56.

Model	Accuracy (%)
ResNet-56	73.71 $\pm$ 0.23
+Randomly MWD	74.27 $\pm$ 0.15
+FocusedDropout without MWD	74.32 $\pm$ 0.11
+Opposite approach without MWD	73.96 $\pm$ 0.08
+Opposite approach	74.31 $\pm$ 0.14
+FocusedDropout	**74.97 $\pm$ 0.16**

Although the network's performance is still improved compared with the baseline, it is not as good as FocusedDropout. We consider that using the opposite method cannot make the network more focused on the target, and it is similar to SpatialDropout, which achieves the effect of overcoming overfitting. Our method makes the network focus more on the target and also can prevent overfitting. This comparative test verifies that FocusedDropout improves network performance by enhancing network attention rather than the randomness.

6. Conclusions

In this work, we introduce a novel regularization method named FocusedDropout. Inspired by Network Dissection, we find that the high activation value units of CNN often

correspond to the classification target, so FocusedDropout first chooses the channel with the highest average activation value as the reference channel and finds the preferred area from it, then only retain the units in this area for all channels due to the spatial invariance of CNN. Extensive experiments prove that FocusedDropout brings stable improvement to different datasets on various models. Additionally, increasing weight decay when using FocusedDropout can prevent the network from overfitting. The class activation mapping suggests the model can more accurately identify the target's location and contour regularized by FocusedDropout. We consider that FocusedDropout provides a new way to improve dropout: finding a balance between making the network focus on the target and preventing overfitting.

Author Contributions: Conceptualization, X.W., M.L. (Ming Liu) and T.X.; methodology, M.L. (Minghui Liu), T.X.; validation, X.W., T.X., M.Y. and J.D.; formal analysis, M.L. (Minghui Liu), T.X. and J.D.; writing—original draft preparation, J.D.; writing—review and editing, X.C. and M.L. (Minghui Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Science and Technology Program of Quzhou under Grant 2021D007, Grant 2021D008, Grant 2021D015, and Grant 2021D018, as well as the project LGF22G010009.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network.
MWD	Magnifying Weight Decay.
CAM	Class Activation Mapping.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
2. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
3. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
5. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
7. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
8. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
9. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
10. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
11. Tompson, J.; Goroshin, R.; Jain, A.; LeCun, Y.; Bregler, C. Efficient object localization using convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 648–656.
12. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Dropblock: A regularization method for convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; pp. 10727–10737.

13. Keshari, R.; Singh, R.; Vatsa, M. Guided dropout. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4065–4072.
14. Ouyang, Z.; Feng, Y.; He, Z.; Hao, T.; Dai, T.; Xia, S.T. Attentiondrop for Convolutional Neural Networks. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 1342–1347.
15. Wang, H.; Wang, G.; Li, G.; Lin, L. CamDrop: A New Explanation of Dropout and A Guided Regularization Method for Deep Neural Networks. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1141–1149.
16. Pham, H.; Le, Q. Autodropout: Learning dropout patterns to regularize deep networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 9351–9359.
17. Bau, D.; Zhou, B.; Khosla, A.; Oliva, A.; Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6541–6549.
18. Krogh, A.; Hertz, J.A. A simple weight decay can improve generalization. In Proceedings of the Advances in neural information processing systems, Denver, CO, USA, 2–5 December 1991; pp. 950–957.
19. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; Citeseer: Princeton, NJ, USA, 2009.
20. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
21. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556
22. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146
23. Öztürk, Ş.; Özkaya, U.; Barstuğan, M. Classification of Coronavirus (COVID-19) from X-ray and CT images using shrunken features. *Int. J. Imaging Syst. Technol.* **2021**, *31*, 5–15. [CrossRef] [PubMed]
24. Shorten, C.; Khoshgoftaar, T.M.; Furht, B. Text data augmentation for deep learning. *J. Big Data* **2021**, *8*, 101. [CrossRef] [PubMed]
25. Peng, Z.; Zhou, J.; Fang, X.; Yan, P.; Shan, H.; Wang, G.; Xu, X.G.; Pei, X. Data Augmentation for Training Deep Neural Networks. In *Auto-Segmentation for Radiation Oncology*; CRC Press: Boca Raton, FL, USA, 2021; pp. 151–164.
26. Wan, L.; Zeiler, M.; Zhang, S.; Le Cun, Y.; Fergus, R. Regularization of neural networks using dropconnect. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 17–19 June 2013; pp. 1058–1066.
27. Kang, G.; Li, J.; Tao, D. Shakeout: A new regularized deep neural network training scheme. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
28. Gal, Y.; Hron, J.; Kendall, A. Concrete dropout. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 3581–3590.
29. Ba, J.; Frey, B. Adaptive dropout for training deep neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3084–3092.
30. Kingma, D.P.; Salimans, T.; Welling, M. Variational dropout and the local reparameterization trick. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2575–2583.
31. Huang, Y.; Sun, X.; Lu, M.; Xu, M. Channel-max, channel-drop and stochastic max-pooling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 9–17.
32. DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552
33. Huang, G.; Sun, Y.; Liu, Z.; Sedra, D.; Weinberger, K.Q. Deep networks with stochastic depth. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 646–661.
34. Gastaldi, X. Shake-shake regularization. *arXiv* **2017**, arXiv:1705.07485.
35. Yamada, Y.; Iwamura, M.; Akiba, T.; Kise, K. Shakedrop regularization for deep residual learning. *arXiv* **2018**, arXiv:1802.02375.
36. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
37. Park, S.; Kwak, N. Analysis on the dropout effect in convolutional neural networks. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; pp. 189–204.
38. Gomez, A.N.; Zhang, I.; Swersky, K.; Gal, Y.; Hinton, G.E. Targeted dropout. In Proceedings of the Neural Information Processing Systems, Montreal, QC, Canada, 2–8 December 2018.
39. Hou, S.; Wang, Z. Weighted channel dropout for regularization of deep convolutional neural network. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January– 1 February 2019.
40. Wu, L.; Li, J.; Wang, Y.; Meng, Q.; Qin, T.; Chen, W.; Zhang, M.; Liu, T.Y. R-drop: Regularized dropout for neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Honolulu, HI, USA, 27 January–1 February 2021; Volume 34, pp. 10890–10905.
41. Fan, X.; Zhang, S.; Tanwisuth, K.; Qian, X.; Zhou, M. Contextual dropout: An efficient sample-dependent dropout module. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
42. Shu, J.; Li, Y.; Wang, S.; Xi, B.; Ma, J. Disease gene prediction with privileged information and heteroscedastic dropout. *Bioinformatics* **2021**, *37*, i410–i417. [CrossRef] [PubMed]
43. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.

44. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
45. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
46. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
47. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.