



A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain

Imrus Salehin  and Dae-Ki Kang * 

Department of Computer Engineering, Dongseo University, 47 Jurye-ro, Sasang-gu, Busan 47011, Republic of Korea; imrus15-8978@diu.edu.bd

* Correspondence: dkkang@dongseo.ac.kr; Tel.: +82-51-320-1724

Abstract: Dropout is one of the most popular regularization methods in the scholarly domain for preventing a neural network model from overfitting in the training phase. Developing an effective dropout regularization technique that complies with the model architecture is crucial in deep learning-related tasks because various neural network architectures have been proposed, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), and they have exhibited reasonable performance in their specialized areas. In this paper, we provide a comprehensive and novel review of the state-of-the-art (SOTA) in dropout regularization. We explain various dropout methods, from standard random dropout to AutoDrop dropout (from the original to the advanced), and also discuss their performance and experimental capabilities. This paper provides a summary of the latest research on various dropout regularization techniques for achieving improved performance through “Internal Structure Changes”, “Data Augmentation”, and “Input Information”. We can see that proper regularization with respect to structural constraints of network architecture is a critical factor to facilitate overfitting avoidance. We discuss the strengths and limitations of the methods presented in this work, which can serve as valuable references for future research and the development of new approaches. We also pay attention to the scholarly domain in the discussion in order to meet the overwhelming increase of scientific research outcomes by providing an analysis of several important academic scholarly issues of neural networks.

Keywords: dropout; regularization; overfitting; data augmentation; dropout benefits



Citation: Salehin, I.; Kang, D.-K. A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. *Electronics* **2023**, *12*, 3106. <https://doi.org/10.3390/electronics12143106>

Academic Editor: Fabio Grandi

Received: 2 June 2023

Revised: 5 July 2023

Accepted: 13 July 2023

Published: 17 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, deep learning techniques have performed remarkably in artificial intelligence improvement and automation. One of the most critical concerns in deep learning research is the regularization method used to prevent overfitting during training. Dropout [1], which was proposed in 2012, is a well-known regularization method that outperforms other regularization processes during optimization. It is used in various neural network tasks, such as image segmentation, data augmentation, image generation, machine translation, and image classification to reduce overfitting during training. The concept of dropout is relatively easy to implement compared to other regularization techniques. Initially, this technique was primarily applied to fully connected models where all neurons in one layer are connected to all neurons in the next layer. However, over time, numerous variations have been implemented for different perspectives and solutions to various problems. Those variations also include a number of modifications to encourage the regularization performances of existing neural network architectures. For example, standard dropout [2], DropConnect [3], DropBlock [4], Skipout [5], MaxDropout [6], DropPath [7], selective CNN dropout, cutout [8], adversarial dropout methods [9], and spectral dropout [10] have attempted to explore the regularization issue from several aspects.

This review paper provides an overview of previous and ongoing research on dropout regularization methods. While not exhaustive, we delve into specific dropout methods

that are particularly relevant or beneficial for specific fields of research. We collect major dropout methods that are widely used in deep feedforward neural networks as well as convolutional neural networks [2,11–22] due to their efficiency and effectiveness. Furthermore, our paper presents a comprehensive review, where we thoroughly investigate and categorize the significant domains of deep learning dropout taxonomies, including convolutional neural networks (CNNs), feedforward neural networks (FNNs), recurrent neural networks (RNNs), and transformers. Recent research has highlighted the increasing popularity and successful application of dropout techniques in recurrent neural networks [4,8–10,23–28] for achieving impressive results in natural language processing tasks. These findings have been widely acknowledged in the state-of-the-art (SOTA) [26,29–37].

The motivation behind the review on dropout techniques in deep learning is to improve model generalization and stability while effectively addressing the issue of overfitting. Dropout methods offer regularization strategies that can improve the robustness of deep learning models by reducing their sensitivity to input variations and noise. By conducting a comprehensive review, researchers can assess the performance trade-offs between different dropout techniques, enabling them to identify the most effective approaches based on the specific requirements of their applications. Furthermore, a thorough review provides valuable insights into the current state-of-the-art, guiding future research endeavors in developing advanced regularization and data augmentation methodologies. Ultimately, the review aims to contribute to the continual improvement of the deep learning model's performance and reliability across various domains and applications.

Our full paper covers various sections, addressing different aspects of dropout regularization. Here is a brief summary of each section: Section 2: Background of regularization and dropout regularization. This section provides an introduction to regularization techniques and specifically focuses on dropout regularization. Section 3: Theoretical concepts. This section explores the theoretical foundations of dropout regularization, discussing its mathematical formulation and underlying principles. Section 4: Contribution of dropout regularization in deep learning. This section examines the benefits and contributions of dropout regularization in the context of deep learning. Sections 5–7: Categorization of dropout regularization and taxonomy-based dropout methods. These sections explore different variations and approaches of dropout regularization, categorizing them based on specific criteria. Sections 10 and 11: Performance comparison with benchmark datasets. These sections present a comparison of the performances of dropout methods using benchmark datasets to assess their effectiveness. Sections 12 and 13: Limitations and open issues. These sections discuss the limitations and potential challenges of dropout methods as well as open research questions and areas for future exploration. Section 14: Conclusion. This section summarizes the main findings and contributions of the paper, emphasizing the efficacy and potential of dropout regularization.

2. Background

In neural networks, regularization is a modification technique to the learning algorithm, which reduces its generalization error with the balance of its training error. This technique is particularly useful in neural networks, where the primary goal is to minimize overfitting during training. The benefits and drawbacks of each technique vary, depending on the specific problem and dataset. In this section, we specifically focus on regularization, with a particular emphasis on dropout regularization. To ensure a comprehensive explanation, we divide this discussion into two subsections.

2.1. Regularization Method

Basic regularization methods are L1 and L2 regularization methods [38]. L1 regularization applies a constraint proportional to the sum of the absolute values of weights while L2 regularization applies a constraint proportional to the sum of the squares of weights. By doing so, the model is encouraged to learn smaller weight values, which in turn helps to prevent overfitting. In recent years, regularization algorithms for deep learning models,

like feedforward and convolutional neural networks, have developed quickly. Extended methods of L1 and L2 regularization include weight decay [39], early stopping [40], and entropy regularization [41].

One popular regularization method is early stopping. This approach monitors the validation loss during the training process and stops it early before the model overfits the training data [40]. The final model is selected based on the lowest validation loss. Batch normalization [42,43] is a method that normalizes the input to each layer of a neural network. It reduces the internal covariate shift and improves the convergence rate by smoothing the optimization landscape. It can also serve as a type of regularization by adding noise to the input of each layer. Data augmentation [44] is a technique used to increase the size of the training dataset by creating modified versions of the original data. The methods of creating modified versions include flipping, rotating, cropping images, etc.

The original dropout [1] method randomly drops out units during the training phase. Dropout methods as regularization algorithms have seen rapid development in deep learning models, like feedforward neural networks (FNNs) and convolutional neural networks (CNNs).

2.2. Regularization with Dropout

Deep neural networks with more trainable parameters are generally desirable for extracting robust features from data. The networks with a large number of trainable parameters are prone to overfitting. To address this issue, in the standard dropout, the network switches off randomly chosen neurons during training [1]. On the one hand, EDropout, an energy-based dropout method, has been employed in [45] to perform the dropout of visible and hidden units in deep neural networks. EDropout introduces an energy-based measure to identify neurons that contribute the least to the overall energy of the network. These neurons are removed during training and testing to reduce the network's complexity and improve its efficiency.

One possible way to enhance the performance and robustness of a CNN model is through the regularization dropout technique. The max-pooling dropout [46] technique randomly drops out some of the max-pooled values during the training of the convolution network. In terms of performance, max-pooling dropout is similar to the original dropout, which selects activation neurons at random [46]. To obtain accurate and robust models before testing, the researchers suggested using probabilistic weighted pooling instead of max-pooling for each feature map. The use of probabilistic weighted pooling has been shown to be effective based on empirical analysis.

In the field of language modeling, Merity et al. introduced several regularization techniques, including dropout, to analyze their impact on LSTM (long short-term memory) language models [47]. In their study, the authors explored different regularization techniques including weight-dropped LSTM, along with different optimization methods, such as Adagrad and Adam. Through their experimentation, they aimed to analyze the impacts of these techniques on LSTM language models and determine their effectiveness in improving model performance.

3. Theoretical Concept of Dropout

In recent times, deep learning architectures have significantly evolved and expanded leading to improvements in various tasks, such as classification, object detection, and segmentation, etc. But, the utilization of larger and more complex deep learning architectures comes with the downside of increased risks of overfitting during training. In order to tackle the challenges in deep learning, researchers have put forward various regularization techniques with dropout being particularly prominent among them.

The standard dropout [1] method can be used to relieve the overfitting issue, which randomly drops neurons from the neural network during training.

For example, consider a neural network with an individual hidden linear layer of N units. Then the activation function Softmax is identical, taking the geometric arithmetic mean of

the final output of the 2^N workable networks under standard Softmax [2]. This method is closer to a machine learning approach, such as the bagging method [48], which has trained the instance separately and the output inference has used the arithmetic mean.

Let us consider a single linear layer within a neural network. This layer is commonly referred to as a linear layer because it incorporates the behavior of a linear activation function, $f(x) = x$. In this (Figure 1) layer, the final neuron (which serves as the output of the layer) is obtained by calculating the weighted sum of all the inputs. While this simplified mathematical explanation holds empirically for certain non-linear networks, it is important to note that the estimation of the model involves minimizing a loss function. The ordinary least square (OLS) loss [49],

$$L_n = \frac{1}{2} \left(t - \sum_{i=1}^n w_i I_i \right)^2 \quad (1)$$

$$L_D = \frac{1}{2} \left(t - \sum_{i=1}^n \delta_i w_i I_i \right)^2 \quad (2)$$

Equation (1) represents the loss function used in general neural networks, while Equation (2) corresponds to the loss function specific to dropout networks. In the dropout network, the dropout rate is denoted by δ , which follows a Bernoulli distribution with parameter p . This implies that δ takes value 1 with probability p and 0 otherwise. In the given network, the input is denoted by “ I ” and the weight associated with each input is represented by “ w ”.

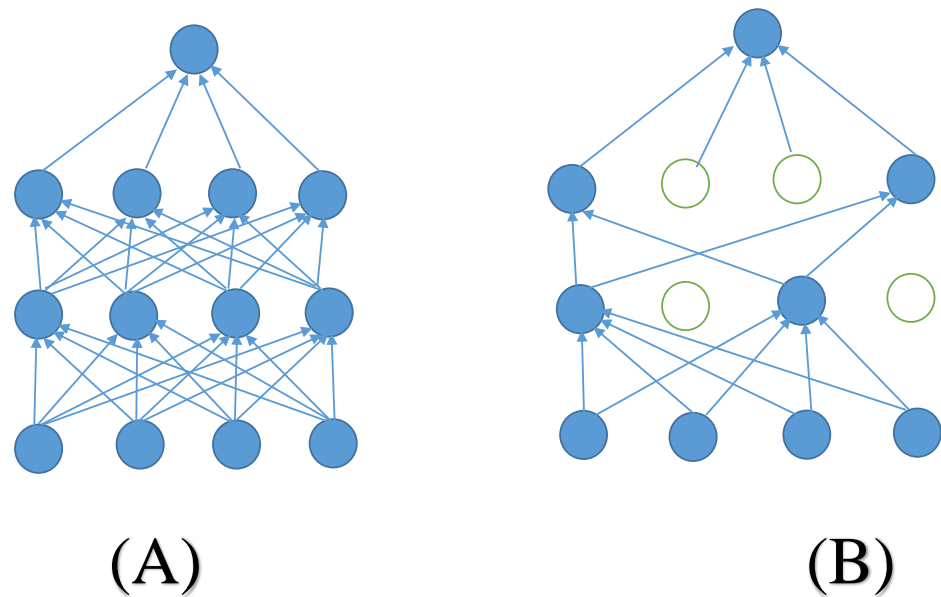


Figure 1. An illustration of a standard neural network (A) and after applying dropout (B).

During network training, the gradient descent approach is utilized for backpropagation. This results in the computation of the gradient of the dropout network, denoted by Equation (2), which subsequently feeds into the regular network, as shown in Equation (1).

$$\frac{dL_D}{dw_i} = -t\delta_i I_i + w_i \delta_i^2 I_i^2 + \sum_{j=1, j \neq i}^n w_j \delta_i \delta_j I_i I_j \quad (3)$$

Now, a relationship can be established between the gradient of the dropout network and the gradient of a regular network. Based on Equation (1), we can assume that $w' = p \times w$, where p represents the probability dropout variable. Therefore, this equation indicates that the weights in the regular network are scaled by the dropout probability, p .

$$L_D = \frac{1}{2} \left(t - \sum_{i=1}^n p_i w_i I_i \right)^2 \quad (4)$$

Taking the derivative of Equation (4),

$$\frac{dL_D}{dw_i} = -t p_i I_i + w_i p_i^2 I_i^2 + \sum_{j=1, j \neq i}^n w_j p_j p_i I_i I_j \quad (5)$$

Now, move on to the next step. When we calculate the expectation of the gradient for the dropout network, we obtain the following expression:

$$\begin{aligned} L \left[\frac{dL_D}{dw_i} \right] &= -t p_i I_i + w_i p_i^2 I_i^2 + w_i \text{Var}(\delta_i) I_i^2 + \sum_{j=1, j \neq i}^n w_j p_j p_i I_i I_j \\ &= \frac{dL_N}{dw_i} + w_i \text{Var}(\delta_i) I_i^2 \\ &= \frac{dL_N}{dw_i} + w_i p_i (1 - p_i) I_i^2 \end{aligned} \quad (6)$$

According to Equation (6), the expectation of the gradient with dropout is equivalent to the gradient of the regular neural network L_N when the weights are scaled by p , denoted as $w' = p \times w$.

4. Contribution of Dropout Regularization Methods

In this section, we discuss the contribution of several commonly used dropout implements based on their performance under the dropout operation. The advantages of dropout regularization methods play a vital role in consolidating knowledge, highlighting key benefits and inspiring further research in the field of deep learning. The primary aim of this section is to highlight the advantages of dropout regularization methods and provide a comprehensive and consolidated overview of the benefits. In the following section, we provide a concise overview of the benefits that these methods offer.

4.1. Effectiveness Improving

The use of dropout techniques has significantly improved the effectiveness of models in utilizing training and promoting data across various procedures.

- **Data augmentation.** These methods [8,41,50,51] are employed to augment the available data. This involves introducing noise to the input data, generating additional training samples and enhancing the effectiveness of the model during training. In each training epoch, a technique called dropout is employed to introduce a new dataset by randomly dropping certain training data.
- **Preventing overfitting.** The majority of dropout approaches [52,53] have been employed for regularization to mitigate overfitting. During the training phase, certain techniques are used to drop out some of the neurons in the model, which can lead to a reduction in the interdependence between preventing overfitting and neurons. This is because the dropout technique randomly drops some neurons during training, making it difficult to establish a clear relationship between individual neurons and the prevention of overlearning. Additionally, dropout has been commonly utilized in deep neural networks to improve their generalization performance.
- **Enhancing data representation.** These methodologies [54–57] are used for enhancing data representation in the pre-training stage. Some methods generate random masks

that are applied to segments of the input data. The unmasked portions of the data are used to predict the masked portion, which helps in increasing the data representation.

- **Preventing over-smoothing.** According to some research analysis, the use of dropout methods in graph convolutional networks has been shown to prevent the over-smoothing problem [58]. When the graph convolutional networks (GCNs) become too deep, this can lead to an over-smoothing problem. This problem causes the representation of every node on the graph to become identical, resulting in a loss of information. Several research studies suggest that dropout methods can be effective in mitigating the over-smoothing issue that can occur with increasingly deep graph convolutional networks (GCNs).

4.2. Efficiency Improving

Dropout techniques can improve both the effectiveness and efficiency of a model across several processes.

- **Model Compression.** Several dropout methods have been utilized for model compression purposes [25,59,60]. By using these methods, the model structure can be made easier to compress after the random dropout of neurons, such as by performing neural pruning. Model summarization methods are used to reduce the number of model parameters, which can lead to improved training efficiency and reduced overfitting.
- **Model uncertainty estimation.** Several well-known techniques have been used to estimate the model uncertainty [15–17,61], such as the authors' interpreting dropout as Bayesian learning in Monte Carlo dropout. Viewing dropout as a Bayesian learning process is a key aspect of these methods. For example, the authors of Monte Carlo dropout interpret dropout as a Bayesian approximation of a deep Gaussian process in their approach. Monte Carlo dropout works with grid search, which is almost unusable in deeper models and reinforcement learning due to the high computational time and resource consumption required.
- **Accelerate GCN training.** In a graph convolutional network, the node feature information sampling process has been proposed by GraphSAGE (sample and aggregate) [62], which efficiently accelerates GCN training. The training purpose requires only some neighbor nodes to execute the training procedure. After conducting our work, we observed similarities between our approach and the FastGCN [63] and AS-GCN [64] methods.

5. Categorization of Dropout Regularization Approaches

During the period from 2012 to 2022, we categorized each dropout regularization approach based on its year of introduction and development. The timeline of these dropout regularization approaches is visually depicted in Figure 2. The graphical representation depicts the temporal continuity of dropout regularization techniques, providing a clear overview of their scenario over the years.

Figure 3 illustrates the taxonomy of dropout approaches, categorized into three main categories: internal structure change, data augmentation, and input information. Each category further consists of several subcategories. By organizing the dropout approaches in this taxonomy, we aim to provide a clear and systematic overview of the different types of dropout techniques based on the changes they induce in the internal structure, their usage in data augmentation, and their impact on input information.

This taxonomy serves as a valuable reference for understanding the various dropout approaches employed in deep learning. The following section provides an in-depth analysis of the taxonomic classification.

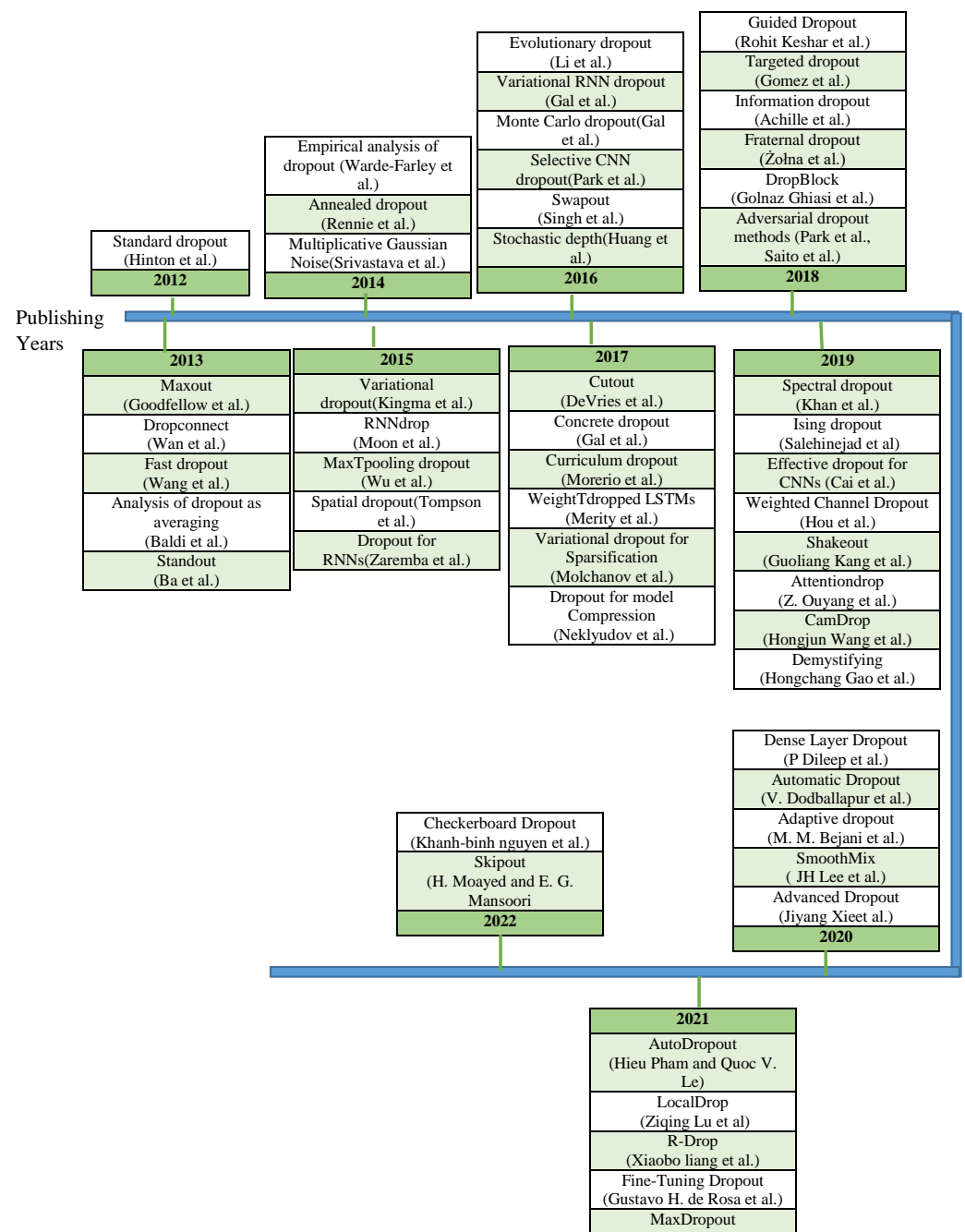


Figure 2. A flow diagram illustrating the progression of dropout methods over the years [2–22].

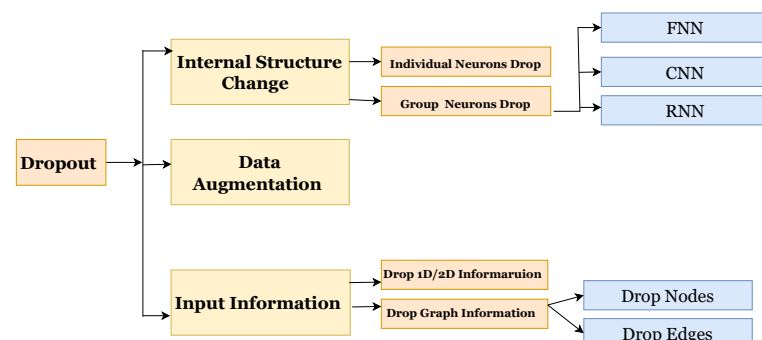


Figure 3. Taxonomy diagram according to dropout approaches.

6. Dropout Based on Internal Structure Changes

Different approaches have been proposed in addition to dropout for regularization in deep learning. In our review paper, we defined internal dropout-based regularization, which involves changing the weights and kernels during the training period without making any changes to the input values.

Individual neuron dropout [1,22,65], as mentioned earlier, involves randomly dropping or deactivating individual neurons during the training phase. Each neuron has a probability of being dropped out, which means its output is set to zero. This helps prevent overfitting by reducing the reliance on specific neurons and encourages the network to learn more robust and generalizable representations.

On the other hand, group neuron dropout takes a different approach. Instead of dropping individual neurons, it drops entire groups or subsets of neurons simultaneously. These groups can be predefined based on various factors, such as their spatial location or functional role within the network. In the following subsection, we provide a detailed discussion of some popular approaches.

6.1. FNN and CNN-Based Neurons Drop Approaches

In this section, we explore various popular neuron dropout approaches based on both feedforward neural networks (FNNs) and convolutional neural networks (CNNs).

6.1.1. DropBlock

Deep neural networks generally achieve high performance when their parameters are over-optimized and trained with significant amounts of noise and regularization techniques like weight decay and dropout. In the course of this research, a method known as DropBlock [4] is a form of a structured dropout that involves removing all units located within a continuous region of a feature map (tensor). The size of the dropped regions (Figure 4) is determined by a hyperparameter, and during each training iteration, the dropped regions are randomly selected.

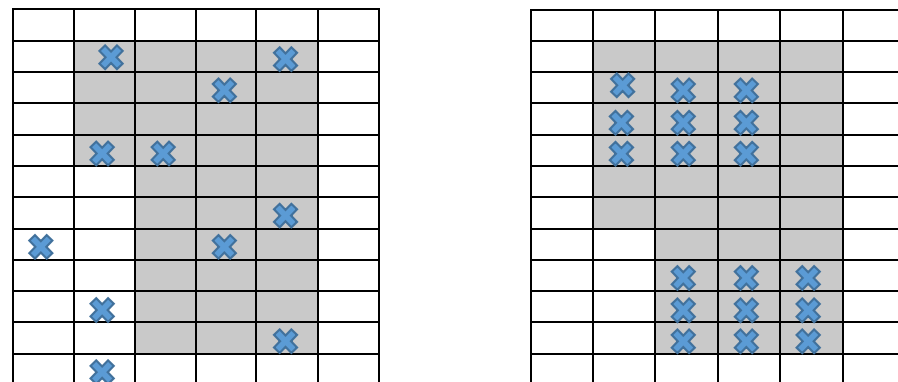


Figure 4. DropBlock structure visualization.

Moreover, gradually increasing the number of units discarded during the training process leads to greater accuracy and makes the model less sensitive to variations in hyperparameters, such as the learning rate and batch size. The image classification task using AmoebaNet-B and ResNet-50 models has been shown to improve the accuracy of the model compared to other variations of dropout. Based on the experimental results, using dropout techniques such as AmoebaNet-B and ResNet-50 can improve the accuracy of image classification tasks compared to other dropout variations. Specifically, on the ImageNet dataset, the baseline accuracy of 2% for ResNet-50 was beaten by using cutout and AutoAugment and AmoebaNet-B achieved an improvement of around 0.3% accuracy.

6.1.2. MaxDropout

While the standard dropout [1] randomly drops the neuron from the training process, MaxDropout [6] deactivated the neuron's calculation from their activations. At the beginning of the experiment, the tensor values are first normalized. After that, they set a 0 value for every single output and mention the thresholds p , so they pick the higher value and deactivate the neuron.

The outcome looks like this in Figure 5. The authors conducted an experiment, where they tested the accuracy of ResNet18 on the CIFAR-10 and CIFAR-100 datasets. The results showed that their method outperformed dropout when applied to the WideResNet-28-10 model [66].

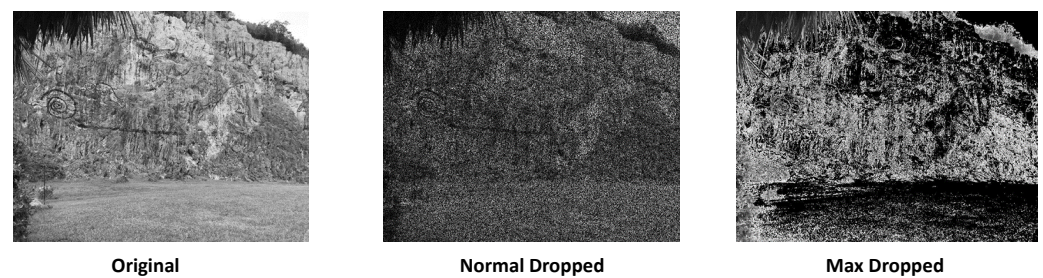


Figure 5. This is a visualization comparing the effects of normal dropout and MaxDropout on an original image.

6.1.3. AutoDrop Dropout

Notwithstanding the effectiveness, the dropout approach lacks information on the dropped neurons in the training phase. In most of the cases and approaches, the neuron dropped randomly. If we think about DropBlock's [4] strategy, they dropped entire random regions in hidden layers and, thus, trained a CNN to acquire knowledge of better spatial information for improved accuracy. AutoDrop dropout [67] is a method of adaptive dropout, where the dropout rate is dynamically adjusted during training based on the activations of the neurons. This allows the model to focus on the most important features during training, while still preventing overfitting. Experimental results have shown that AutoDrop dropout can improve the accuracy of image classification tasks and language modeling compared to traditional dropout methods. It has also been shown to be effective in other types of neural networks, such as transformer models.

6.1.4. AttentionDrop

The TargetDrop [68] drops the feature before coming up with a clear target. This approach is an advanced version of DropBlock and it dropped the target regions in the consisting feature map based on mask. An attention mechanism could be included in a specified regularized so that it could take action in the congruous region. This approach first selects the mask using the activation value then fixes the shape of the mask and randomly generates the adaptive mask to address the issue of lacking information on dropped neurons in the training phase. The experimental results demonstrate that AttentionDrop enhances the performance of convolutional neural networks on public benchmark datasets, specifically, SVHN. AttentionDrop is an effective dropout-based regularization method used across various scales of networks. According to results visualized using Grad-CAM [69], this method performs better than the accomplished DropBlock.

6.2. Dropout with Recurrent Neural Networks (RNNs)

One of the main challenges of applying dropout to RNN is that RNN has recurrent connections, which means that the same weights are used at each time step. This can cause problems with dropout because dropping out of different units at each time step can result in inconsistent and unstable behavior, which can harm the performance of the RNN. Yarin Gal and Zoubin Ghahramani illustrated a robust technique for applying

dropout regularization to RNN [15]. The key idea of this paper is to show how dropout regularization can be applied to RNN in a theoretically grounded manner. The authors derived a mathematical formulation for dropout in RNN based on the notion of adding noise to hidden units. V. Pham. et al. [70] demonstrate how dropout regularization can improve the performance of RNN models for handwriting recognition tasks. They applied dropout only to feedforward connections and not to recurrent connections. In practice, they implemented dropout as a separate layer that outputs the same values as the input, except at the dropped locations. “Recurrent dropout” [71] represents another dropout with RNN without losing important information from the hidden state. This approach introduces a “masking” tensor that is applied to the hidden state before it is passed through the activation function. Tensor is randomly generated for each training iteration with probabilities determined by a dropout rate hyperparameter.

7. Dropout Based on Data Augmentation

Data augmentation is the process of modifying data to reduce variance by making slight modifications to images and is used to generate more training data. Some dropout methods can serve as a form of data augmentation allowing for improved generalization without requiring specific domain knowledge [50]. In this section, we discuss the effectiveness of dropout based on data augmentation as a technique for improving model performance. While focusing on high-performance dropout based on data augmentation, we explore its capabilities and demonstrate its effectiveness in augmenting data for improved model performance.

7.1. Cutout

This method is a straightforward but potential technique for data augmentation [8]. During the training process, cutout serves as a technique for image augmentation and regularization. It involves masking out square regions of the input data individually.

In [8], the researcher analyzed the optimal sizes for the dropped part in the CIFAR-100 and CIFAR-10 datasets. According to the experiment, the optimal size varied depending on the number of instances in each class, which was calculated from the given dataset. Furthermore, this cutout has been used to enhance the strength and accomplishment performances of convolutional neural networks. For the best results, the CIFAR-10 dataset was processed by dropping a region of size 16×16 (height and width) while for CIFAR-100, the region size yielding the best results was 8×8 . (Figure 6) illustrates the working process of the cutout.

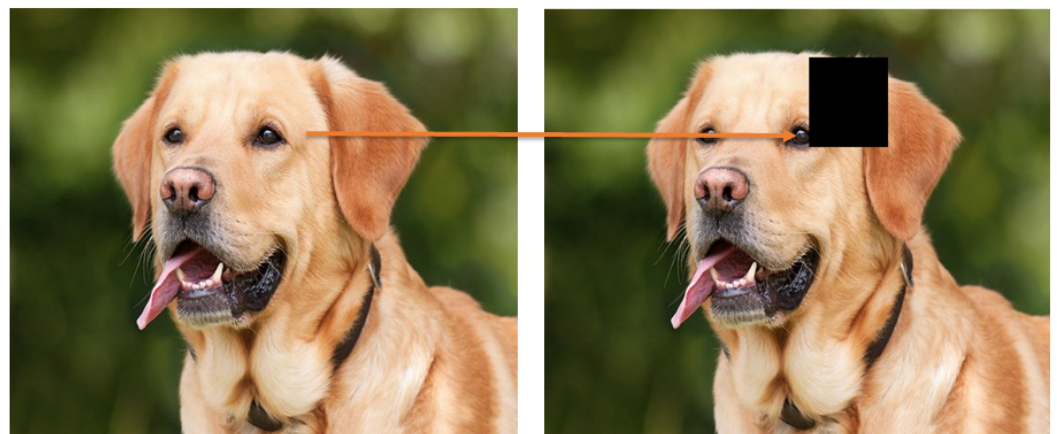


Figure 6. This figure illustrates the work process of the cutout.

7.2. Random Erasing

Random erasing is a process of data augmentation, which is one of the most commonly and comprehensively used method (Figure 7). This method operates in a similar fashion to dropout [51]. This process randomly selects an area of an image and replaces the pixels

in that area with random values, effectively “erasing” that portion of the image. In convolutional neural networks, it has been used in training times to reduce overfitting and it enhances the robustness of a model. For the best performance in this method, several architectures on Fashion-MNIST, CIFAR-10, and CIFAR-100 have been used. But, this mechanism works well for the WRN-28-10 and ResNeXt-8-64 models compared to other architecture models.

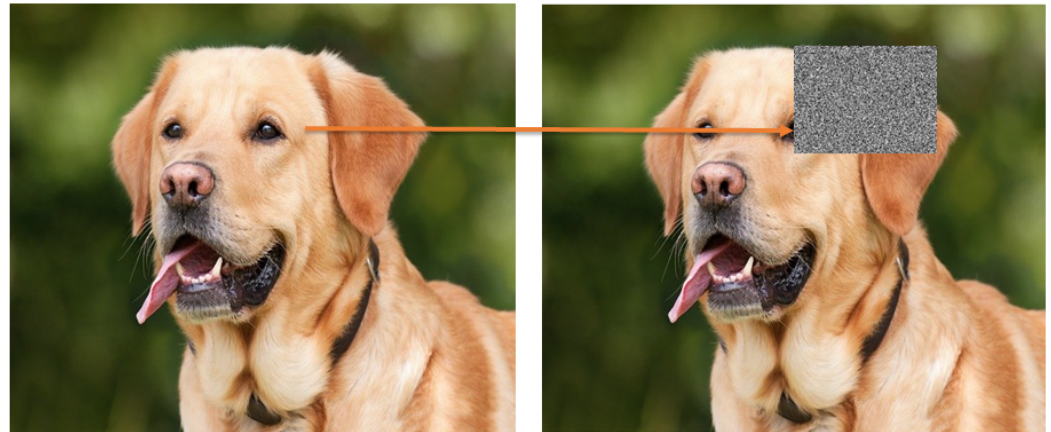


Figure 7. This figure illustrates the work process of random erasing.

According to the gray-scale image analysis, random erasing improves the performance of the WRN-28-10 architecture from 4.01% to 3.65% in the top-1% error rate on the benchmark dataset, Fashion-MNIST.

7.3. Demystifying Dropout

Among several dropout and regularization techniques, demystifying dropout [33] offers a different perspective on dropout models. This method has been developed by theoretical and mathematical concepts and involves forward and backward passes. It improves the performances of both standard dropout [1] and data augmentation [8]. The key point in demystifying dropout is that it randomly drops features during the forward pass, but it does not drop any features or backpropagate errors during the backward pass. On the other hand, in backward dropout, all features are kept in the forward pass, even when a feature is dropped, and only the gradients of the dropped features are set to zero in the backward pass. In this study, the authors have presented a novel perspective on augmented dropout.

7.4. SmoothMix

SmoothMix is a state-of-the-art regularization method that helps prevent overfitting and improve the performance of data augmentation [72]. They proposed minimizing the strong-edge problem by applying a smooth change of the boundary between two images. In this technique, authors have introduced kernel filters and injected noise to create an augmented input between two images. For performance evaluation, they compared the experimental results using ResNet-50 and Pyramid Net-200 architecture models on ImageNet and CIFAR-100 datasets.

8. Dropout Based on Input Information

Some dropout methods directly discard a portion of input information during training, serving various purposes in different scenarios. These purposes may include regularization, data augmentation, or enhancing data representation during the pretraining stage. In 2016, Sennrich et al. [73] introduced WordDropout as a technique in machine translation, which involves dropping out words from the input data. Another notable technique, proposed by Ghazvinine et al. [74] in 2019, is Mask-Predict. While most machine translation systems

generate text from left to right, Mask-Predict adopts a masking approach to train the model. It begins by predicting all target words and then iteratively masking and regenerating a subset of words for which the model exhibits the least confidence.

Current data enhancement methods can be broadly categorized into three main groups: spatial transformation, color distortion, and information dropping. Within the realm of information dropping, the act of discarding two-dimensional input information is commonly considered a data enhancement technique. In 2019, Verma et al. [75] introduced manifold mixup as a novel approach. Manifold mixup extends the concept of mixup to the feature level. The underlying idea is that features contain higher-order semantic information, and performing interpolation at the feature level can yield more meaningful samples. By applying manifold mixup, the model can benefit from the diverse and informative characteristics of different features. In 2019, Yun et al. [76] introduced CutMix as an advancement over the techniques of mixup and cutout. While cutout fills specific regions of an image with meaningless content, which may hinder the model's ability to utilize the complete training data, mixup employs linear interpolation but generates images that may not resemble natural images. In contrast, CutMix addresses these limitations by randomly selecting rectangular regions from one image (X_a) and replacing them with corresponding regions from another image (X_b) at the same locations.

The graph neural networks approach provides a more natural and informative augmentation by blending different image regions. Graph neural networks (GNNs) find extensive application in diverse tasks, like node classification, cluster detection, and recommender systems. When training GNNs, certain methods employ the random dropout of nodes or edges, utilizing only a subset of graph information for training [77,78]. This dropout technique serves as a regularization mechanism to prevent overfitting and enhances the generalization ability of the model. Following the development of GraphSAGE, several node-dropout training methods have emerged. In 2018, Chen et al. introduced FastGCN, which shares similarities with GraphSAGE [64]. The key distinction lies in the FastGCN approach of randomly sampling nodes from the entire graph, rather than solely focusing on the neighbors of a specific node. This broader sampling strategy significantly enhances the efficiency of node sampling. As a result, FastGCN outperforms the original GCN and GraphSAGE in terms of computational speed while still maintaining comparable prediction performance.

9. Dropout with Transformer

In deep learning, the transformer is a type of neural network architecture that was introduced by Vaswani [79] in 2017. Transformers are designed for sequential input data, such as natural language sentences or time series data. Transformers leverage a self-attention mechanism that enables them to process the input sequence in parallel, unlike conventional recurrent neural networks (RNN) that process data sequentially. A. Fan et al. [80] introduced a new type of dropout technique called "structured dropout", which allows for the efficient reduction in the number of layers in a transformer model at the inference time. At inference time, the structured dropout technique is used to drop the non-important layers according to the learned mask. This allows for the efficient reduction in the depth of the transformer model, which can save computation time and memory. Wu et al. [81] proposed a cutting-edge transformer-based dropout regularization method named UniDrop, which amalgamates three dropout modalities: feature dropout, structure dropout, and data dropout. PatchDropout [82] is a dropout regularization technique designed for vision transformer models. Unlike traditional dropout techniques that drop out individual tokens or features, PatchDropout randomly drops out patches of the input image during training. This reduces the computational and memory requirements of vision transformers while improving their generalization and robustness.

10. Materials for Experiments

In our review paper, we provide a direct comparison of each dropout method by analyzing the experimental results and performance. In this section, we provide a discussion of the dataset and a basic explanation of the experimental architecture. To facilitate a more transparent comparison, we divided our experimental materials into parallel subsections.

10.1. Datasets

The training phase is one of the most crucial parts of a neural network. It relies heavily on both the baseline dataset and the data that are used for training, evaluation, and validation. These data sources are critical for ensuring the accuracy and effectiveness of the network's performance. The experiment on the dropout method utilized a majority of the benchmark dataset to showcase the algorithm's performance on various benchmark datasets, including CIFAR-10, CIFAR-100, and ImageNet.

CIFAR (Canadian Institute For Advanced Research) is a well-renowned dataset, which consists of 80 M images. However, for this experiment, most researchers have used two other subsets frequently: (i) CIFAR-10 and (ii) CIFAR-100.

The *CIFAR-10* subset is made with 60,000 image instances, each with a size of 32×32 ; it is divided into 50,000 instances for training and the remaining 10,000 instances for testing and validation. This dataset is compounded into 10 classes, representing various animals and objects.

The *CIFAR-100* subset is made with 60,000 image instances, each with a size of 32×32 ; it is divided into 50,000 instances for training and the remaining 10,000 instances for testing and validation. However, this dataset is compounded into 100 classes, representing animals and objects, which are classified; this is the counterpart-updated version.

The *ImageNet* dataset is developed by the ImageNet project, which is developed for artificial intelligence tasks, such as image classification, image detection, etc. This dataset has been introduced from the "2012 ImageNet Large Scale Visual Recognition Challenge" (ILSVRC). This dataset comprises 1,240,000 instances with 224×224 images for training and 50,000 for testing/validation purposes. It contains 1000 classes, which are bigger than the CIFAR subsets.

10.2. Architectures

ResNet. The residual network is a CNN architecture, which is the oldest architecture for regularization works. This ResNet [83] group is one of the most commonly used architectures for convolutional neural networks. Basically, this ResNet uses residual connections, and concatenates the output layer from the previous layer with further transformations.

Different experiments use two variants of ResNet architecture; these network variants are different because this depends on the depth of the neural network. The ResNet-18 [8] and ResNet-50 [4] are the most popular subvariants of the ResNet family. ResNet-18 is built on a 72-layer architecture as well as 18 deep layers. This architecture trains with the ImageNet database for the convolutional neural network with more than a million images. On the other hand, ResNet-50 consists of 50 layers; it is trained on the ImageNet database.

ViT-B/16. The ViT [84] models focus on the vision transformer. The visual transformer partitions an image into fixed-patch sizes; after that, it embeds each of them, including a positional embedding system. In CNN, the ViT model has shown superior performance in terms of both computational efficiency and accuracy.

Wide residual network—WRN. The WRN [66] is another comprehensive architecture for regularization works. The concept of this architecture is the same concept as that of ResNet but the residual connection between layers and some structural differences have been established.

PyramidNet. The most common neural network architecture is PyramidNet [85], as mentioned in our experiment table, and is widely used for regularization analysis. This model has the lid out and 2D faces that can be folded to make the 3D shape of a pyramid. PyramidNet increases the feature map in the network layer and moderately increases the

dimensionality. This procedure shows the improvements of the classification task results in the neural networks.

11. Performance Comparison with Dropout Experiment Results

Convolutional neural networks (CNNs) and transformers are generally designed to perform the best optimal outcome in image processing. Dropout is an effective regularization method that can improve the performance and accuracy of neural networks by reducing overfitting and improving the smoothness of the model's output. Regularization methods, along with other components, can be easily incorporated into further experiments. Table 1 presents the results of individual CNN and transformer architectures on the CIFAR-100, CIFAR-10, and ImageNet datasets.

Table 1. Error % for each classification dataset using different methods and models.

Method	Classification Datasets		ImageNet
	CIFAR-10	CIFAR-100	
ResNet-50 [4]	-	-	23.49 ± 0.07
+ AutoDrop + RA [67]	-	-	19.7
+ Skipout [5]	4.66	20.61	-
+ AutoDrop [67]	-	-	21.3
+ RA [86]	-	-	22.4
+ FixRes [87]	-	-	17.5
+ DropBlock [4]	-	-	21.65 ± 0.05
+ Mixup [87]	-	-	22.1
+ Fast AA [88]	-	-	22.37
+ BA [89]	-	-	23.14
+ AA [4]	-	-	22.4
+ Cutout [4]	-	-	23.48 ± 0.07
+ Dropout [4]	-	-	23.20 ± 0.04
+ LocalDrop [90]	5.3	26.2	21.1
+ ShakeDrop [91]	-	25.26	-
+ Bag of Tricks [92]	-	-	21.67
+ GradAug [93]	-	-	20.33
ViT-B/16 [84]	-	7.36	16.03
+ ViT-B/16 + RD [94]	-	6.71	15.62
+ ViT-L/16 + RD [94]	-	6.15	14.43
ResNet18 [8]	4.72 ± 0.21	22.46 ± 0.31	-
+ Cutout [8]	3.99 ± 0.13	21.96 ± 0.24	-
+ MaxDropout [6]	4.66 ± 0.14	21.93 ± 0.07	-
+ MaxDropout + Cutout [6]	3.76 ± 0.08	21.82 ± 0.13	-
+ TargetDrop [95]	4.41	21.37	-
+ TargetDrop + Cutout [95]	3.67	21.25	-
+ LocalDrop [90]	4.3	22.2	-
Wide Residual Networks [66]	4.00	19.25	21.9
+ Dropout [66]	3.89	18.85	-
+ MaxDropout [6]	3.84	18.81	-
+ TargetDrop [95]	4.41	21.37	-
+ TargetDrop + Cutout [8]	3.08 ± 0.16	18.41 ± 0.27	-
+ AutoDrop [67]	3.1	-	-
+ AutoDrop + RE [67]	2.1	-	-
+ Dropout + RA [86]	2.7	16.7	-
PyramidNet [85]	3.48 ± 0.20	17.01 ± 0.39	19.2
+ GradAug [93]	-	13.76	20.94
+ ShakeDrop + RA [86]	1.5	-	15.0

This is particularly advantageous for datasets with limited training samples, as small datasets often face challenges related to overfitting. Dropout helps alleviate this issue by preventing models from memorizing the training data and instead encourages the

learning of generalizable patterns. Moreover, dropout regularization proves beneficial in datasets with significant noise or variability. By randomly dropping out units during training, dropout regularization enables models to capture more robust features that are less influenced by noise, ultimately enhancing their performance on such datasets. In addition, dropout regularization is effective in high-dimensional datasets where the number of features surpasses the number of training samples. The curse of dimensionality is a common obstacle in high-dimensional datasets, making it difficult for models to generalize effectively.

The above section provides an overview and in-depth analysis of the experiments considered in this paper. The following section attempts to identify the strengths and limitations of various dropout models based on their performance and implementation. We also provide programming language information, and the GitHub source code can be found in Table A1 of our Appendix A section.

12. Strengths and Limitations

In this section, we focus on recent and major dropout approaches, highlight their strengths, and attempt to identify their limitations based on previous research in Table 2. Some dropout approaches have extended their methods to overcome limitations while others still work within these limitations to improve their effectiveness.

Table 2. Evaluation of strengths and limitations.

Approach	Strengths	Limitations
Dropout [1]	It reduces the overfitting in the training layer, including supervised learning, speech recognition, and document classification.	For graph neural networks, convolutional operations and recurrent neural network performances may be relatively lower.
DropBlock [4]	Creates a structure for dropped neurons.	The structure works as a fixed structure and randomly creates this structure.
AutoDrop [67]	It has an automatic pattern to drop the neuron.	This method demonstrated improved performance for transformer and text-processing analyses only.
Cutout [8]	No need for domain knowledge to generate the mask.	For the complex architecture and noise image dataset, the performances were not that high.
R-Drop [94]	Careful about dropped unit inference during the training stage.	Only outperform transformer and text-processing analyses according to performance results.
Fraternal dropout [28]	Reduces the gap between sub-models and minimizes the loss function.	Inconsistency between training and inference.
ELD [96]	Regularizes the gap between sub-models during inference.	It may require more computational resources compared to standard dropout
Skipout [5]	Skipout dropout does not require tuning of additional hyperparameters, making it easy to implement and use in practice.	Layer-by-layer training approach is critical during training for complex architectures.
DropGNN [97]	Smooths the performance of any graph neural network by reducing its complexity.	It can lead to a significant loss of information when dropping entire nodes or edges in the graph, which can negatively impact the model's performance.
MaxDropout [6]	It can inactivate neurons based on the maximum activation values from the network.	Only works for image classification tasks.
LocalDrop [90]	It works for complexity analyses of both fully connected layers and convolutional neural networks.	LocalDrop has mathematical- and parameter-based dependencies, which make it time-consuming.
Advanced dropout [98]	It provides insights into the importance of specific regions or features in the input data, making it useful for the interpretability and understanding of complex models.	Suffers from the convergence of the adaptive dropout rate in a series of analyses.

In Table 3, we demonstrate a quick summary of several dropout approaches based on their key characteristics. In the beginning, we will consider domain knowledge, which means the understanding of the specific problem domain in which dropout is applied. This includes knowledge of the data being used, the architecture of the neural network, and the specific objectives of the task. We also consider whether the automatic neuron is active or inactive in a neural network. Lastly, we investigate the process of the dropped neurons. The exploration between random and non-random dropout patterns have been identified. In practice, this approach mainly focuses on preserving or discarding important information about neurons.

Table 3. Quick summary of several dropout approaches

Approach	Domain Knowledge	Automatic	Random Dropped
Dropout [1]	No	No	Yes
DropBlock [4]	Yes	Yes	Yes
AutoDrop [67]	Yes	Yes	No
Cutout [8]	No	No	Yes
R-Drop [94]	Yes	No	Yes
Fraternal dropout [28]	Yes	No	Yes
ELD [96]	Partial	No	Yes
Skipout [5]	Partial	No	Yes
DropGNN [97]	Partial	No	Yes
MaxDropout [6]	No	No	No
DropConnect [3]	Yes	Yes	Yes

13. Discussion and Open Issue

In our study, we conducted a comprehensive examination of the well-known regularization dropout method, which is a popular and effective technique for improving training convergence and final performance compared to other regularization approaches. Our performance table compares the performances of various dropout methods and baseline architectures using different benchmark datasets.

As illustrated in Table 1, the ResNet-50 architecture with the ImageNet dataset exhibits superior performance when implementing AutoDrop + RA and FixRes regularization techniques, as evidenced by the lower error percentage compared to other methods. For the ResNet18 architecture trained on the CIFAR-10 dataset, similar results were observed, where it showed better performance compared to other methods, such as MaxDropout+cutout and cutout. The dropout technique is effective at reducing overfitting and speeding up training for deep neural networks with large and complex computational depths.

In Figure 8, we present an overview of the overall characteristics of dropout methods [1,3–6,28,67,94,96,97] in a statistical view. Chart 1 specifically focuses on the percentages of dropout methods that meet the three main criteria: domain knowledge, automatic implementation, and random dropping. This chart provides insights into the extent to which dropout methods incorporate these criteria. Additionally, Chart 2 depicts the statistics for dropout methods that do not fulfill all three criteria. This chart highlights the prevalence of dropout methods that may lack one or more of the specified characteristics. By utilizing these two charts, we provide a comprehensive understanding of the distribution and prevalence of dropout methods based on their adherence to the defined criteria. By analyzing both Charts 1 and 2, we can observe that a significant majority of dropout methods possess domain knowledge, indicating that they are based on prior understanding or expertise in a specific domain. It is worth noting that the dropout approaches depicted in these charts lack automatic implementation meaning that the dropout process is not inherently automated. Furthermore, the dropout rate in these methods is typically determined by the user rather than being predefined or automatically optimized. This indicates that users have the flexibility to adjust the dropout rate according to their specific requirements or preferences. By varying the dropout rate, users can regulate the amount of regularization applied to the network, thereby influencing the trade-off between model complexity and generalization.

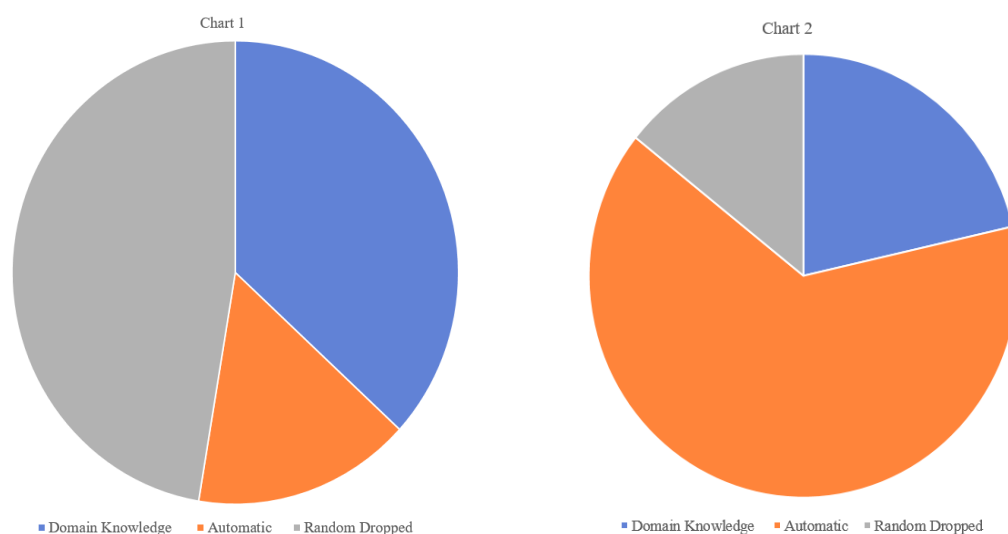


Figure 8. Statistical overview of dropout based on characteristics.

An open issue in dropout regularization involves the selection of an optimal dropout rate, which requires balancing model capacity and overfitting prevention. One potential solution is to explore adaptive dropout techniques that dynamically adjust dropout rates based on the network's needs during training. Another open issue is the application of dropout in sequential models, which can be addressed by developing specialized dropout variants tailored for sequential data. Additionally, improving the robustness of dropout in adversarial settings is an open challenge; potential solutions include investigating dropout variants specifically designed to enhance model resilience against adversarial attacks. Unbalanced datasets, where the distribution of classes is highly skewed, pose a challenge for traditional machine learning approaches. Dropout regularization has the potential to address this issue by preventing the overfitting of the majority class and promoting learning from the minority class. There is a need to investigate the effectiveness of dropout in handling class imbalance scenarios and explore strategies for adapting dropout to better handle imbalanced data, such as class-dependent dropout rates or adjusting dropout rates during training. Dropout regularization has primarily been investigated and applied to the context of static data, such as images or tabular data. When it comes to sequential and temporal data, such as time series or natural language processing tasks, incorporating dropout poses unique challenges.

Still, ongoing research is being conducted to improve the dropout approach for better performance in trained neural networks. Many researchers are currently working on reducing overfitting in neural networks by improving the dropout method. Specialized methods have been developed for neural networks, convolutional neural networks, and recurrent neural networks to address the limitations of the standard dropout method in practice. As mentioned above in the dropout experiment, specialized dropout methods, such as adversarial dropout [9,16,99] have been investigated.

The scholarly field of AI encompasses the ongoing development of machine learning and deep learning as well as all related scientific research aimed at advancing and benefiting our society. The utilization of these methods can result in enhancements in the automation of software, tools, and systems for scientific research outcomes in the scholarly domain.

14. Conclusions

In this study, we observed more than forty regularization methods based on dropout, which can be categorized into several major groups (i.e., feedforward, convolutional, recurrent neural network, and transformer) with dropout operation and performance. Throughout our discussion, we explored various strategies, applications, interconnections, and contributions of these methods in the realm of neural networks. It is important to

note that the selection of the most appropriate dropout method depends on the specific problem and architecture, and further research is needed to explore their full potential. The utilization of dropout in deep learning models has proven to be remarkably effective in enhancing performance and addressing overfitting.

By randomly deactivating a portion of neurons during training, dropout encourages the network to learn more robust and generalized features, ultimately leading to improved performance on unseen test data. This ability enhances model generalization and effectively mitigates the risk of overfitting, resulting in more robust and reliable neural networks. With its simplicity and effectiveness, dropout has become a widely adopted technique in the deep learning community, empowering researchers and practitioners to build more reliable and high-performing neural networks. In order to make the robustness of dropouts stronger in the future, researchers can delve into the exploration and development according to our review discussion. This capability enhances the generalization of models and effectively mitigates the risk of overfitting, leading to more robust and reliable neural networks. With its simplicity and effectiveness, dropout has gained widespread adoption in the deep learning community, empowering researchers and practitioners to construct more reliable and high-performing neural networks. To further strengthen the robustness of dropout techniques in the future, researchers can delve into various avenues for exploration and development, as discussed in our review. By advancing our understanding of dropout methods and their applications, we can continue to enhance the performance and reliability of deep learning models.

Author Contributions: Conceptualization, I.S. and D.-K.K.; data curation, I.S.; formal analysis, I.S. and D.-K.K.; funding acquisition, D.-K.K.; investigation, I.S.; methodology, I.S.; project administration, D.-K.K.; software, I.S.; supervision, D.-K.K.; validation, D.-K.K.; visualization, I.S.; writing—original draft, I.S.; writing—review and editing, D.-K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2022R1A2C2012243).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors wish to thank the members of the Dongseo University Machine Learning/Deep Learning Research Lab, as well as the anonymous reviewers for their helpful comments on earlier drafts of this paper.

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

Abbreviations

The following abbreviations are used in this manuscript:

AA	AutoAugment
BA	BatchAugmentation
CREX	CRedible EXplanation Regularizing Approach
DNN	Deep Neural Networks
Fast AA	Fast AutoAugment
GAN	Generative Adversarial Networks
HACDB	Handwritten Arabic Characters Database
IFN/ENIT	Database Handwritten Arabic Characters
MM	ManifoldMixup
PBA	Population-Based Augmentation
RE	RandomErasing
RA	RandAugment

Appendix A

In this section, we have included Table A1, which provides information about the programming language used for implementing the dropout technique and the corresponding GitHub resource.

Table A1. The source code links (coding language and GitHub) for dropout.

Short Name	Source Code	Accessed Date
Dropout [2]	https://pytorch.org/docs/stable/_modules/torch/nn/modules/dropout.html PyTorch	20 December 2022
CutMix [76]	https://github.com/clovaai/CutMix-PyTorch PyTorch, GitHub	22 December 2022
Cutout [8]	https://github.com/uoguelph-mlrg/Cutout PyTorch, GitHub	5 January 2023
MaxDropout [6]	https://github.com/cfsantos/MaxDropout-torch/ PyTorch, GitHub	10 January 2023
DropBlock [4]	https://github.com/tensorflow/tpu/tree/master/models/official/resnet TensorFlow, GitHub	10 January 2023
Curriculum dropout [24]	https://github.com/pmorerio/curriculum-dropout TensorFlow, GitHub	10 January 2023
ShakeDrop [91]	https://github.com/imenurok/ShakeDrop Torch, GitHub	15 January 2023
Fraternal dropout [28]	https://github.com/kondiz/fraternal-dropout Python, GitHub	17 January 2023
DropGNN [77]	https://github.com/KarolisMart/DropGNN Python, GitHub	17 January 2023
SmoothMix [72]	https://github.com/jh-jeong/smoothmix Roff, GitHub	18 January 2023
RandomErasing [51]	https://github.com/zhunzhong07/Random-Erasing PyTorch, GitHub	21 January 2023
Mixup [75]	https://github.com/facebookresearch/mixup-cifar10 Python, GitHub	21 January 2023

References

- Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
- 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.
- Wan, L.; Zeiler, M.; Zhang, S.; Le Cun, Y.; Fergus, R. Regularization of Neural Networks using DropConnect. In *Proceedings of Machine Learning Research, Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2023*; Dasgupta, S., McAllester, D., Eds.; PMLR: Atlanta, GA, USA, 2013; Volume 28, pp. 1058–1066.
- Ghiasi, G.; Lin, T.Y.; Le, Q.V. Dropblock: A regularization method for convolutional networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 10750–10760.
- Moayed, H.; Mansoori, E.G. Skipout: An Adaptive Layer-Level Regularization Framework for Deep Neural Networks. *IEEE Access* **2022**, *10*, 62391–62401. [\[CrossRef\]](#)
- Do Santos, C.F.G.; Colombo, D.; Roder, M.; Papa, J.P. MaxDropout: Deep neural network regularization based on maximum output values. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milano, Italy, 10–15 January 2021*; pp. 2671–2676.
- Larsson, G.; Maire, M.; Shakhnarovich, G. FractalNet: Ultra-Deep Neural Networks without Residuals. *arXiv* **2016**, arXiv:1605.07648.
- DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552.
- Park, S.; Park, J.; Shin, S.J.; Moon, I.C. Adversarial dropout for supervised and semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence, Orleans, LA, USA, 2–7 February 2018*; Volume 32.
- Khan, S.H.; Hayat, M.; Porikli, F. Regularization of deep neural networks with spectral dropout. *Neural Netw.* **2019**, *110*, 82–90. [\[CrossRef\]](#)
- Poernomo, A.; Kang, D.K. Biased dropout and crossmap dropout: Learning towards effective dropout regularization in convolutional neural network. *Neural Netw.* **2018**, *104*, 60–67. [\[CrossRef\]](#)
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
- Warde-Farley, D.; Goodfellow, I.J.; Courville, A.; Bengio, Y. An empirical analysis of dropout in piecewise linear networks. *arXiv* **2013**, arXiv:1312.6197.
- Rennie, S.J.; Goel, V.; Thomas, S. Annealed dropout training of deep networks. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT), South Lake Tahoe, NV, USA, 7–10 December 2014*; pp. 159–164.
- Gal, Y.; Ghahramani, Z. A theoretically grounded application of dropout in recurrent neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1027–1035.
- Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016*; pp. 1050–1059.

17. 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; Springer: Berlin/Heidelberg, Germany, 2016; pp. 646–661.
18. Kingma, D.P.; Salimans, T.; Welling, M. Variational dropout and the local reparameterization trick. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2573–2575.
19. Moon, T.; Choi, H.; Lee, H.; Song, I. RNNDROP: A novel dropout for RNNS in ASR. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Scottsdale, AZ, USA, 13–17 December 2015; pp. 65–70.
20. 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.
21. Ba, J.; Frey, B. Adaptive dropout for training deep neural networks. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3084–3092.
22. Li, Z.; Gong, B.; Yang, T. Improved dropout for shallow and deep learning. In *Advances in Neural Information Processing Systems, Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 29.
23. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent neural network regularization. *arXiv* **2014**, arXiv:1409.2329.
24. Morerio, P.; Cavazza, J.; Volpi, R.; Vidal, R.; Murino, V. Curriculum Dropout. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3544–3552.
25. Molchanov, D.; Ashukha, A.; Vetrov, D. Variational dropout sparsifies deep neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 2498–2507.
26. Lindmar, J.H.; Gao, C.; Liu, S.C. Intrinsic sparse LSTM using structured targeted dropout for efficient hardware inference. In Proceedings of the IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Incheon, Republic of Korea, 13–15 June 2022; pp. 126–129.
27. Achille, A.; Soatto, S. Information dropout: Learning optimal representations through noisy computation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2897–2905.
28. Zolna, K.; Arpit, D.; Suhubdy, D.; Bengio, Y. Fraternal dropout. *arXiv* **2017**, arXiv:1711.00066.
29. Salehinejad, H.; Valaee, S. Ising-dropout: A regularization method for training and compression of deep neural networks. In Proceedings of the ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 3602–3606.
30. Cai, S.; Shu, Y.; Chen, G.; Ooi, B.C.; Wang, W.; Zhang, M. Effective and efficient dropout for deep convolutional neural networks. *arXiv* **2019**, arXiv:1904.03392.
31. 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; Volume 33, pp. 8425–8432.
32. 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.
33. Gao, H.; Pei, J.; Huang, H. Demystifying Dropout. In Proceedings of the 36th International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 10–15 June 2019; Volume 97, pp. 2112–2121.
34. Dileep, P.; Das, D.; Bora, P.K. Dense layer dropout based CNN architecture for automatic modulation classification. In Proceedings of the National Conference on Communications (NCC), IEEE, Kharagpur, India, 21–23 February 2020; pp. 1–5.
35. Dodbballapur, V.; Calisa, R.; Song, Y.; Cai, W. Automatic Dropout for Deep Neural Networks. In Proceedings of the International Conference on Neural Information Processing, Vancouver, BC, Canada, 6–12 December 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 185–196.
36. Bejani, M.M.; Ghaate, M. Adaptive low-rank factorization to regularize shallow and deep neural networks. *arXiv* **2020**, arXiv:2005.01995.
37. Ahmed, R.; Gogate, M.; Tahir, A.; Dashtipour, K.; Al-Tamimi, B.; Hawalah, A.; El-Affendi, M.A.; Hussain, A. Novel Deep Convolutional Neural Network-based Contextual Recognition of Arabic Handwritten Scripts. *Entropy* **2021**, *23*, 340.
38. Ng, A.Y. Feature selection, L1 vs. L2 regularization, and rotational invariance. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; p. 78.
39. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
40. Liu, H.; Brock, A.; Simonyan, K.; Le, Q. Evolving normalization-activation layers. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 13539–13550.
41. Chen, P.; Liu, S.; Zhao, H.; Jia, J. Gridmask data augmentation. *arXiv* **2020**, arXiv:2001.04086.
42. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 448–456.
43. Santurkar, S.; Tsipras, D.; Ilyas, A.; Madry, A. How Does Batch Normalization Help Optimization? In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18, Montreal, QC, Canada, 3–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 2488–2498.
44. Perez, L.; Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621.
45. Salehinejad, H.; Valaee, S. Edropout: Energy-based dropout and pruning of deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 5279–5292. [[CrossRef](#)]

46. Wu, H.; Gu, X. Max-pooling dropout for regularization of convolutional neural networks. In Proceedings of the Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, 9–12 November 2015; Proceedings, Part I 22; Springer: Berlin/Heidelberg, Germany, 2015; pp. 46–54.
47. Merity, S.; Keskar, N.S.; Socher, R. Regularizing and optimizing LSTM language models. *arXiv* **2017**, arXiv:1708.02182.
48. Opitz, D.; Maclin, R. Popular Ensemble Methods: An Empirical Study. *J. Artif. Intell. Res.* **1999**, *11*, 169–198. [[CrossRef](#)]
49. Baldi, P.; Sadowski, P.J. Understanding dropout. In *Advances in Neural Information Processing Systems 26, Proceedings of the NIPS 2013, Lake Tahoe, NV, USA, 5–10 December 2013*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 26.
50. Bouthillier, X.; Konda, K.; Vincent, P.; Memisevic, R. Dropout as data augmentation. *arXiv* **2015**, arXiv:1506.08700.
51. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13001–13008.
52. Wager, S.; Wang, S.; Liang, P.S. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems, Proceedings of the NIPS'13: 26th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 5–10 December 2013*; Curran Associates Inc.: Red Hook, NY, USA, 2013; Volume 26.
53. Helmbold, D.P.; Long, P.M. On the inductive bias of dropout. *J. Mach. Learn. Res.* **2015**, *16*, 3403–3454.
54. Mianjy, P.; Arora, R. On Dropout and Nuclear Norm Regularization. In Proceedings of the 36th PMLR 2019 International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Proceedings of Machine Learning Research Series; Chaudhuri, K., Salakhutdinov, R., Eds.; Volume 97, pp. 4575–4584.
55. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H. Ernie: Enhanced representation through knowledge integration. *arXiv* **2019**, arXiv:1904.09223.
56. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 13–18 July 2020; pp. 11328–11339.
57. Zhou, K.; Wang, H.; Zhao, W.X.; Zhu, Y.; Wang, S.; Zhang, F.; Wang, Z.; Wen, J.R. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; pp. 1893–1902.
58. Rong, Y.; Huang, W.; Xu, T.; Huang, J. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv* **2019**, arXiv:1907.10903.
59. Gomez, A.N.; Zhang, I.; Kamalakara, S.R.; Madaan, D.; Swersky, K.; Gal, Y.; Hinton, G.E. Learning sparse networks using targeted dropout. *arXiv* **2019**, arXiv:1905.13678.
60. Neklyudov, K.; Molchanov, D.; Ashukha, A.; Vetrov, D.P. Structured bayesian pruning via log-normal multiplicative noise. In *Advances in Neural Information Processing Systems, Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 30.
61. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems, Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 30.
62. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems, Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 30.
63. Chen, J.; Ma, T.; Xiao, C. Fastgcn: Fast learning with graph convolutional networks via importance sampling. *arXiv* **2018**, arXiv:1801.10247.
64. Huang, W.; Zhang, T.; Rong, Y.; Huang, J. Adaptive sampling towards fast graph representation learning. In *Advances in Neural Information Processing Systems, Proceedings of the NIPS'18: 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 31.
65. Chen, Y.; Yi, Z. Adaptive sparse dropout: Learning the certainty and uncertainty in deep neural networks. *Neurocomputing* **2021**, *450*, 354–361. [[CrossRef](#)]
66. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
67. Pham, H.; Le, Q. Autodropout: Learning dropout patterns to regularize deep networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 9351–9359.
68. 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.
69. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
70. Pham, V.; Bluche, T.; Kermorvant, C.; Louradour, J. Dropout Improves Recurrent Neural Networks for Handwriting Recognition. In Proceedings of the 2014 14th International Conference on Frontiers in Handwriting Recognition, Hersionissos, Greece, 1–4 September 2014; pp. 285–290.
71. Semeniuta, S.; Severyn, A.; Barth, E. Recurrent Dropout without Memory Loss. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–17 December 2016; The COLING 2016 Organizing Committee: Osaka, Japan, 2016; pp. 1757–1766.

72. Lee, J.H.; Zaheer, M.Z.; Astrid, M.; Lee, S.I. Smoothmix: A simple yet effective data augmentation to train robust classifiers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 756–757.
73. Sennrich, R.; Haddow, B.; Birch, A. Edinburgh Neural Machine Translation Systems for WMT 16. In Proceedings of the First Conference on Machine Translation, Berlin, Germany, 11–12 August 2016; Shared Task Papers; Association for Computational Linguistics: Berlin, Germany, 2016; Volume 2, pp. 371–376.
74. Ghazvininejad, M.; Levy, O.; Liu, Y.; Zettlemoyer, L. Mask-predict: Parallel decoding of conditional masked language models. *arXiv* **2019**, arXiv:1904.09324.
75. Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6438–6447.
76. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6023–6032.
77. Weng, Y.; Chen, X.; Chen, L.; Liu, W. GAIN: Graph attention & interaction network for inductive semi-supervised learning over large-scale graphs. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 4257–4269.
78. Zhao, H.; Yao, Q.; Tu, W. Search to aggregate neighborhood for graph neural network. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 552–563.
79. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems, Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 30.
80. Fan, A.; Grave, E.; Joulin, A. Reducing transformer depth on demand with structured dropout. *arXiv* **2019**, arXiv:1909.11556.
81. Wu, Z.; Wu, L.; Meng, Q.; Xia, Y.; Xie, S.; Qin, T.; Dai, X.; Liu, T.Y. Unidrop: A simple yet effective technique to improve transformer without extra cost. *arXiv* **2021**, arXiv:2104.04946.
82. Liu, Y.; Matsoukas, C.; Strand, F.; Azizpour, H.; Smith, K. PatchDropout: Economizing Vision Transformers Using Patch Dropout. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–7 January 2023; pp. 3942–3951.
83. 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.
84. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
85. Han, D.; Kim, J.; Kim, J. Deep pyramidal residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5927–5935.
86. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 702–703.
87. Liang, D.; Yang, F.; Zhang, T.; Yang, P. Understanding Mixup Training Methods. *IEEE Access* **2018**, *6*, 58774–58783. [[CrossRef](#)]
88. Lim, S.; Kim, I.; Kim, T.; Kim, C.; Kim, S. Fast AutoAugment. In *Advances in Neural Information Processing Systems, Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019*; Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
89. Hoffer, E.; Ben-Nun, T.; Hubara, I.; Giladi, N.; Hoefler, T.; Soudry, D. Augment your batch: Improving generalization through instance repetition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8129–8138.
90. Lu, Z.; Xu, C.; Du, B.; Ishida, T.; Zhang, L.; Sugiyama, M. LocalDrop: A hybrid regularization for deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3590–3601. [[CrossRef](#)] [[PubMed](#)]
91. Yamada, Y.; Iwamura, M.; Akiba, T.; Kise, K. Shakedrop regularization for deep residual learning. *IEEE Access* **2019**, *7*, 186126–186136. [[CrossRef](#)]
92. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 558–567.
93. Yang, T.; Zhu, S.; Chen, C. Gradaug: A new regularization method for deep neural networks. In *Advances in Neural Information Processing Systems, Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 33, pp. 14207–14218.
94. 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 *Advances in Neural Information Processing Systems, Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Virtual, 6–14 December 2021*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 34, pp. 10890–10905.
95. Zhu, H.; Zhao, X. TargetDrop: A targeted regularization method for convolutional neural networks. In Proceedings of the ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; pp. 3283–3287.

96. Ma, X.; Gao, Y.; Hu, Z.; Yu, Y.; Deng, Y.; Hovy, E. Dropout with expectation-linear regularization. *arXiv* **2016**, arXiv:1609.08017.
97. Papp, P.A.; Martinkus, K.; Faber, L.; Wattenhofer, R. DropGNN: Random Dropouts Increase the Expressiveness of Graph Neural Networks. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 21997–22009.
98. Xie, J.; Ma, Z.; Lei, J.; Zhang, G.; Xue, J.H.; Tan, Z.H.; Guo, J. Advanced dropout: A model-free methodology for bayesian dropout optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 4605–4625. [[CrossRef](#)]
99. Park, S.; Song, K.; Ji, M.; Lee, W.; Moon, I.C. Adversarial Dropout for Recurrent Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4699–4706.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.