electronics

MDPI

Article

Two-Branch Attention Learning for Fine-Grained Class Incremental Learning

Jiaqi Guo [1], Guanqiu Qi [2], Shuiqing Xie [1] and Xiangyuan Li [1,*]

[1] College of Electronics and Information Engineering, South-Central University for Nationalities, Wuhan 430074, China; 201821111413@mail.scuec.edu.cn (J.G.); xieshuiqing@mail.scuec.edu.cn (S.X.)
[2] Computer Information Systems Department, State University of New York at Buffalo State, Buffalo, NY 14222, USA; qig@buffalostate.edu
* Correspondence: lixiangyuan@mail.scuec.edu.cn

Abstract: As a long-standing research area, class incremental learning (CIL) aims to effectively learn a unified classifier along with the growth of the number of classes. Due to the small inter-class variances and large intra-class variances, fine-grained visual categorization (FGVC) as a challenging visual task has not attracted enough attention in CIL. Therefore, the localization of critical regions specialized for fine-grained object recognition plays a crucial role in FGVC. Additionally, it is important to learn fine-grained features from critical regions in fine-grained CIL for the recognition of new object classes. This paper designs a network architecture named two-branch attention learning network (TBAL-Net) for fine-grained CIL. TBAL-Net can localize critical regions and learn fine-grained feature representation by a lightweight attention module. An effective training framework is proposed for fine-grained CIL by integrating TBAL-Net into an effective CIL process. This framework is tested on three popular fine-grained object datasets, including CUB-200-2011, FGVC-Aircraft, and Stanford-Car. The comparative experimental results demonstrate that the proposed framework can achieve the state-of-the-art performance on the three fine-grained object datasets.

Keywords: class incremental learning; fine-grained visual categorization; attention; convolutional neural network

Citation: Guo, J.; Qi, G.; Xie, S.; Li, X. Two-Branch Attention Learning for Fine-Grained Class Incremental Learning. *Electronics* **2021**, *10*, 2987. https://doi.org/10.3390/electronics10232987

Academic Editors: Yanping Zhang, Zhifeng Xiao and Jianjun Yang

Received: 27 October 2021
Accepted: 25 November 2021
Published: 1 December 2021

1. Introduction

In the real world, a visual system may involve constantly emerging new objects. The visual system should be able to keep the recognition performance on existing objects when it keeps learning to recognize new objects [1]. As a straightforward approach of computer vision, pretrained models, such as VGG [2], Inception [3,4] or ResNet [5], are finetuned on a new training dataset for the recognition of new objects. However, this may lead to a common issue—catastrophic forgetting. To be more specific, one pretrained model finetuned on a new dataset result in considerable performance drop on previous datasets. Therefore, class incremental learning (CIL) is proposed to learn a unified classifier for both previous and new object classes. As a major reason, the imbalance between previous and new training data causes catastrophic forgetting [6–8]. Existing CIL methods [9–13] can be divided into three categories: replay-based [9], regularization-based [10,13], and architecture-based [12] methods. In replay-based methods, a tiny exemplar subset of the previous dataset is stored to reduce the forgetting. In [9], samples that are the closest ones to the average sample of each class are selected and added to the tiny exemplar set. However, there is still a large room to improve. As a typical example of regularization-based methods, distillation regularization term, which encourages the outputs of a current model to be similar to the reference model, is introduced into the loss function used in [13]. In [10], several regularization terms such as forgetting-less constraint and inter-class separation are introduced to rebalance the previous and new data. In architecture-based methods, novel architectures are designed to solve existing issues in CIL, such as the stability–plasticity

dilemma [14]. For example, there are two kinds of residual blocks in adaptive aggregation network (AANet) [12]. Specifically, stable and plastic blocks are designed to preserve the previous knowledge and learn new knowledge, respectively.

Although some existing studies focus on CIL, the datasets applied to the experiments, such as CIFAR100 [15] and ImageNet [16], are mostly coarse-grained. In these datasets, there is a wide gap between most of the categories. In other words, the difference between inter-class objects in these datasets is large and easier to be captured by a learning system. However, the CIL for fine-grained objects have received less attention. Fine-grained visual categorization (FGVC) is a more challenging visual task due to the subtle differences between fine-grained subcategories. The primary goal for FGVC methods is to learn effective fine-grained feature representation. There are two general directions to approach this goal. The first is to localize and crop critical regions for the extraction of fine-grained features, and the second is to directly learn effective fine-grained feature representation in an end-to-end fashion. The difference between these two methods mainly is whether or not to intercept local critical regions.

In this paper, we focus on the model's ability to learn fine-grained feature representation under a CIL setting, which has not been extensively studied in prior efforts. Our study strives to answer two questions: (1) How well do existing FGVC models perform in a CIL setting? (2) How can attention mechanism help boost a model's performance via better localization and usage of critical regions for fine-grained CIL? To answer the first question, we adopt a CIL process proposed in [10], which divides the training in multiple incremental phases [17]. Initially, a certain number of fine-grained categories are first used to train a model. Then the obtained model is further trained in the subsequential phases to recognize new fine-grained categories. To prevent catastrophic forgetting, a fixed-sized exemplar set is kept and updated. During the incremental learning, samples from the exemplar set also participate in training to refresh the memory of the model. Existing FGVC models are plugged into this CIL framework for evaluation. To answer the second question, we design a novel neural architecture named two-branch attention learning network (TBAL-Net), which leverages attention modules to better localize critical regions. These highlighted parts are further cropped and fed back into the backbone network to boost feature learning.

In summary, the core contribution of this study is the proposal of TBAL-Net for fine-grained CIL. TBAL-Net focuses on the feature mining of an object's critical regions, which can be effectively learned through an attention mechanism. A series of experiments have been conducted to validate the efficacy of TBAL-Net in a comparison with several baseline models on three widely used FGVC datasets, including CUB-200-2011 [18], FGVC-Aircraft [19], and Stanford Cars [20]. Results demonstrate that TBAL-Net achieves consistent performance gains in the top-1 accuracy compared to its peers. In addition, we have quantitively verified the positive effect brought by the attention module via an ablation study.

The rest of this paper is organized as follows. Section 2 provides a review of relevant studies in CIL and FGVC tasks. Section 3 presents a detailed description of the proposed TBAL-Net model. Section 4 reports the experimental design, results, and analysis. Lastly, we summarize this work in Section 5.

2. Related Work

2.1. Class Incremental Learning

Class incremental learning (CIL) [7] methods aim to learn effective feature representation for both previous and new classes. "Catastrophic forgetting" [21,22] occurs frequently in deep neural network (DNN) as reported in CIL which refers to a degradation of the performance on previous dataset when the model is trained to adapt to a new dataset. Such a bias in performance exists extensively in the CIL approaches currently. In LwF [13], knowledge distillation regularization term is first introduced into the loss function to retain the knowledge learnt from the previous training data. Knowledge distillation refers to distilling knowledge from a model of a cumbersome teacher and infusing it to a model

of a light student which applied extensively in teaching [23–26] and can contribute to a generalization of model [27–29]. As shown in [9], LwF prefers to process new classes in the inference phase. To solve this problem, iCaRL [9] proposes a classification strategy called nearest-mean-of-exemplars. In this strategy, a prototype is computed by averaging the features extracted from all samples of the same class. In the inference phase, the class labels of the most similar prototypes are assigned to testing samples. iCaRL also constructs an exemplar set with the fixed memory size. The samples that are the closest ones to the average prototype of each class are stored in the exemplar set. Although iCaRL has improved the performance of CIL, it still shows a bias to new classes. The main reason is the imbalance between the previous and new classes. To further improve the performance, Ref. [10] introduced several regularization terms such as forgetting-less constraint and inter-class separation to solve the problem of imbalance between previous and new classes. Besides this, Ref. [12] proposed a new network architecture with stable and plastic blocks to deal with the stability–plasticity dilemma in CIL.

2.2. Fine-Grained Visual Categorization

In FGVC, it is important to localize the critical regions for the recognition of fine-grained objects [30–32]. So, localization subnetworks [33–35] are designed in many exiting FGVC methods. In [33], a navigator–teacher–scrutinizer network (NTS-Net) was proposed as a multi-agent learning framework to learn fine-grained features and localize informative regions simultaneously without any bounding-boxes or part annotations. Ref. [34] proposed a network architecture called multi-branch and multi-scale attention learning (MMAL) for the localization of critical regions, which used less parameters than the previous work. In MMAL, a large critical part is first localized and then subtle critical parts are localized with multiple scales. In [35], recurrent attention convolutional neural network (RA-CNN) was proposed to recursively learn discriminative region attention and region-based features at multiple scales.

3. Proposed Method

In this paper, a two-branch attention learning network (TBAL-Net) is designed for the recognition of fine-grained objects in fine-grained CIL. The network first trains in the initial phase and then learns to recognize new fine-grained objects with additional training data. In this section, the architecture of TBAL-Net is introduced in Section 3.1. Then the details of the CIL process applied to the experiments are discussed in Section 3.2.

The overall process of our method can be summarized as follows. TBAL-Net with the backbone CNN pretrained on ImageNet is trained on existing classes in the initial phase. To mitigate catastrophic forgetting, an exemplar set with the fixed memory size is constructed. Samples selected from this exemplar set are the most similar ones to the prototypes of each class. Catastrophic forgetting is addressed in CIL by finetuning the model on this rebalanced exemplar set. To further improve the performance of CIL, distillation regularization, forgetting-less constraint regularization, and inter-class separation regularization are introduced like in [10].

3.1. Network Architecture

The architecture of TBAL-Net is shown in Figure 1. Two attention modules are added to the backbone. The parameters used in the backbone network of TBAL-Net are defined as $P_{backbone}$, and the parameters used in attention modules are defined as $P_{atten,i}$, $i = 1,2 \ldots$, n, where n is the number of attention modules applied to TBAL-Net.

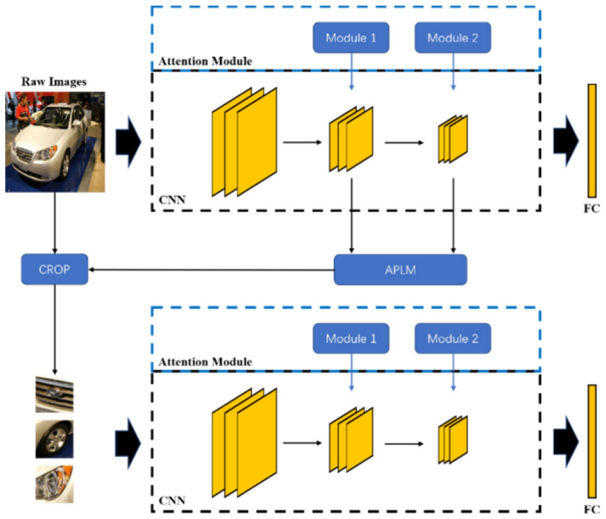


Figure 1. The framework of TBAL-Net. Both channel and spatial attention modules, namely, module 1 and module 2 in the figure, are added to the CNN backbone for feature extraction. The extracted feature maps are fed into the APLM, where critical regions are highlighted and used to guide the cropping module, which crops a number of top-informative regions from the raw image. Finally, the generated part images are fed into the CNN backbone for further feature extraction, aiming to learn more visual patterns from the part images to enhance feature representation for FGVC. The extracted feature map passes a fully connected (FC) layer followed by the detection head.

Attention module. Similar to [36], the attention module also contains channel and spatial attention modules. The feature map is defined as $\boldsymbol{F} \in \boldsymbol{R}^{C \times H \times W}$. In the channel attention module, average and maximum pooling operations are applied to the spatial dimensions. The results of each pooling operation are defined as $\boldsymbol{F}_{avg}^{C} \in \boldsymbol{R}^{C \times 1 \times 1}$ and $\boldsymbol{F}_{max}^{C^{C \times 1 \times 1}}$. Each result of average and maximum pooling operations is first fed into multiple layer perception (MLP). The outputs of MLP are then summarized together. Sigmoid function is applied to summation. In short, the output of the channel attention module is defined as

$$C_{attention} = sigmoid(MLP(AvgPool(\boldsymbol{F})) + MLP(MaxPool(\boldsymbol{F}))) \quad (1)$$

In spatial attention module, average and maximum pooling operations are applied to the channel dimension. The results of each pooling operation are defined as $\boldsymbol{F}_{avg}^{S} \in \boldsymbol{R}^{1 \times H \times W}$ and $\boldsymbol{F}_{avg}^{S} \in \boldsymbol{R}^{1 \times H \times W}$. These two maps are concatenated according to the channel dimension. The result of the feature map is first convolved by a standard convolutional layer. The output of the convolutional layer is then fed into a sigmoid function. In short, the output of the spatial attention module is defined as

$$S_{attention} = sigmoid\left(f^{s \times s}([AvgPool(\boldsymbol{F}); MaxPool(\boldsymbol{F})])\right) \quad (2)$$

In the experiments, channel and spatial attention modules are integrated into a Res-Block, as shown in Figure 2.

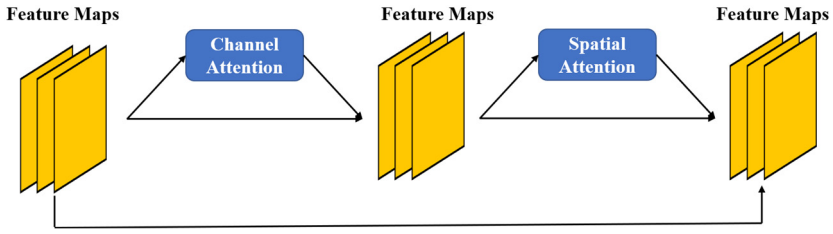


Figure 2. Attention modules with both channel and spatial attention modules similar to [36]. Both attention modules can be added to any location in CNN.

Attention part localization module (APLM). In APLM, activation maps are used to localize the critical regions. Activations in the convolutional layer can be considered as the informativeness of regions with a certain window size. In [34], the activations mean the values are computed according to

$$\bar{a}_w = \frac{\sum_{x=0}^{W_w-1}\sum_{y=0}^{H_w-1} \boldsymbol{F}_w(x,y)}{H_w \times W_w} \tag{3}$$

where H_w and W_w are the height and width of a feature map in a window. As an informativeness measure, activations mean the values of all windows are sorted to localize the most informative regions. To reduce the region redundancy, non-maximum suppression (NMS) is adopted to select a fixed number of windows with different scales. In TBAL-Net, the parameters of backbone CNN and FC layer are shared by both two branches.

The category probabilities of two branches in TBAL-Net are defined as P_r and P_p. The loss function is defined as

$$L_{total} = L_{raw} + L_{parts} \tag{4}$$

where

$$L_{raw} = -log(P_r(c)) \tag{5}$$

$$L_{parts} = -\sum_{i=0}^{N-1} log(P_{p(i)}(c)) \tag{6}$$

3.2. Class Incremental Learning

In this paper, TBAL-Net is integrated into the CIL process introduced in [10]. The data of the previous class C_o is defined as X_o, and the data of a new class C_n is defined as X_n. As shown in Figure 3, CIL can be considered as an $(N+1)$-phase training process, i.e., one initial phase and N incremental phases. In the initial phase, training data X_o is available for training the TBAL-Net parameterized by θ_o. The FC layer of TBAL-Net is initialized as a fully connected layer. After the initial phase, only a small subset of X_o can be stored in an exemplar set with the fixed size. In the following N incremental phases, all samples from the new classes and previously selected exemplar set are first used to train the model. The output of FC layer in TBAL-Net is extended to $|C_o|+|C_n|$.

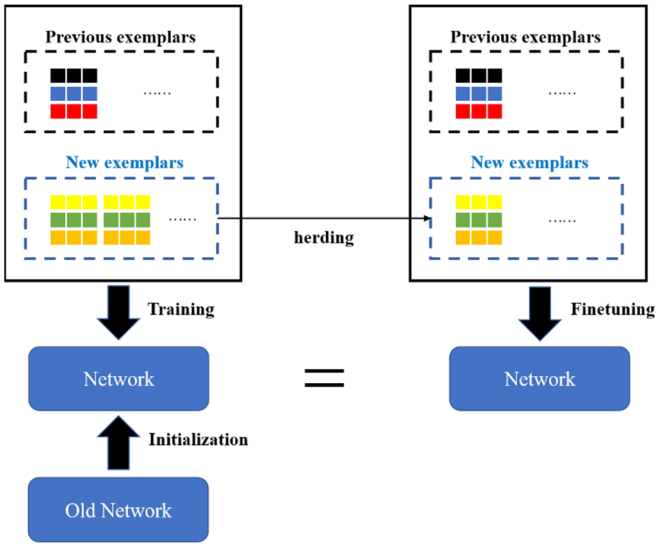


Figure 3. The CIL method used in our experiments. The whole training process consists of $(N + 1)$ phases, including one initial phase and N incremental phases. The initial phase of training takes a subset of data to train an initial model to recognize a subset of classes, while the following phases incrementally add more samples to fine tune the model to recognize new object classes. To prevent catastrophic forgetting, a fixed-sized exemplar set is kept and updated to include typical training samples for all classes that have been seen. At each incremental phase, samples from the exemplar set are also used for training and will be updated to stay current after the incremental learning completes.

To balance magnitudes across all classes, cosine normalization is applied to the last layer of TBAL-Net as follows.

$$p_i(x) = \frac{exp(\lambda \langle \bar{\theta}_i, \bar{f}(x) \rangle)}{\sum_j exp(\lambda \langle \bar{\theta}_j, \bar{f}(x) \rangle)}, \tag{7}$$

where $\bar{v} = v / \left|\left|v\right|\right|_2$ denotes the l_2 normalization vector, $\langle \cdot, \cdot \rangle$ denotes the cosine similarity between two vectors, and λ is a learnable scale parameter, which is introduced to control the peak of softmax distribution, since the cosine similarity is restricted to $[-1, 1]$. λ can be updated through back propagation. Through cosine normalization, all scores before softmax distribution are in the same range and thus are comparable. The distillation loss in [10] is defined as

$$L_{dis}(x) = 1 - \langle \bar{f}^*(x), \bar{f}(x) \rangle, \tag{8}$$

where $\bar{f}^*(x)$ and $\bar{f}(x)$ are two normalized features extracted from the original and current models, respectively. Different from the distillation loss shown in LwF [13], this term encourages the orientation of features extracted by the current network to be similar to those extracted by the original model.

The inter-class separation regularization term is defined as

$$L_{mr}(x) = \sum_{k=1}^{K} max(m - \langle \bar{\theta}(x), \bar{f}(x) \rangle + \langle \bar{\theta}^k, \bar{f}(x) \rangle, 0), \tag{9}$$

The objective of this process is defined as

$$L = \frac{1}{|N|} \sum_{x \in N} (L_{ce}(x) + \pi L_{dis}(x)) + \frac{1}{|N_o|} \sum_{x \in N_o} L_{mr}(x), \tag{10}$$

where L_{ce} is a traditional cross-entropy loss function.

4. Experiments

4.1. Datasets

Experiments in this paper are conducted on three popular fine-grained object datasets, i.e., CUB-200-2011 [18], FGVC-Aircraft [19], and Stanford Cars [20]. The details of these three datasets are introduced in Table 1. In the experiments, only image labels are used without involving any part annotations.

- **CUB-200-2011**. It is the most widely used fine-grained visual categorization dataset. For each subcategory, about 30 images are used for training and 11–30 images for testing.
- **Stanford Car**. In this dataset, each subcategory contains 24–84 images for training and 24–83 images for testing.
- **FGVC-Aircraft**. This dataset is organized into a three-layer label structure. The three layers, from bottom to top, consist of 100 variants, 70 families, and 30 manufacturers, respectively. It is split into 6667 training images and 3333 test images. In the experiments, we considered the case of dividing the images into 70 families.

Table 1. Datasets in the experiments.

Datasets	Number of Classes	Training	Testing
CUB-200-2011	200	5994	5794
FGVC-Aircraft	70	6667	3333
Stanford Cars	196	8144	8041

4.2. Baselines

In the experiments, both traditional CNN such as ResNet50 and several FGVC methods such as NTS-Net and MMAL are evaluated. NTS-Net contains more parameters than MMAL. According to the discussion of Section 3.2, all FC layers in these baselines are extended in the experiments.

- **ResNet50**. As a traditional CNN architecture, ResNet50 [5] pretrained on ImageNet is chosen as a feature extractor. The pretrained FC layer is deleted from the architecture and a new initialized random FC layer is added to the network. Following the experimental setting in [10], when adopting cosine normalization in the last layer, the ReLU in the penultimate layer is removed to allow the features to take both positive and negative values.
- **NTS-Net**. Critical regions with different sizes and aspect ratios are automatically selected by a region proposal network. It could fuse both local and global features for recognition. ResNet50 is the backbone network of NTS-Net. The number of proposal regions is set to 3. In the experiments, the number of learnable parameters in NTS-Net is about 2.8 M. The backbone network in NTS-Net is pretrained on ImageNet dataset. The final feature is obtained through the summation of global and local features. When NTS-Net is trained on the initialized training data, the cosine normalization is also added to the last layer. When facing the new classes, the trainable parameters is added in the FC layer for training.
- **MMAL**. The backbone of MMAL is also ResNet50, which has been pretrained on the ImageNet dataset. In the attention object location module (AOLM), the outputs of Conv5_b and Conv5_c are used for localization of objects. In the attention part proposal module (APPM), the settings of each dataset are same as the settings used in this paper. In the experiments, the number of learnable parameters in MMAL is about 2.6 M. Similar to the setting in NTS-Net, the final feature in MMAL is also obtained through the summation of global and parts features. The trainable parameters in the FC layer, which is shared by the global and local branch, are also added when facing new classes.

4.3. Implementation

In this paper, we apply TBAL-Net to the CIL framework proposed by [10]. All experiments are implemented with PyTorch and trained on a PC with four TITAN-X GPUs.

We adopt the ResNet50 pretrained on ImageNet as the backbone in TBAL-Net. For all three datasets, the learning rate starts from 0.001 and is divided by 10 every 30 epochs (90 epochs in total). The TBAL-Net is trained by SGD with the batch size 32 (8 for each GPU). In the training phase, the input image is first resized to 512×512, and then randomly flipped and cropped the region with a size of 448×448 from the image. In the testing phase, the input image is first resized to 512×512 and then cropped the region with the size of 448×448 from the image center. The part images are resized to 224×224, three broad categories of scales: $\{[4 \times 4, 3 \times 5], [6 \times 6, 5 \times 7], [8 \times 8, 6 \times 10, 7 \times 9, 7 \times 10]\}$ are construct for feature map of 14×14. The number of a raw image's part images is $N = 7$, among them $N_1 = 2$, $N_2 = 3, N_3 = 2$. The number of parts in TBAL-Net are set to be the same as in [34]. The reason for doing this is that the selected regions based on the activations are basically stable. Moreover, the candidate regions processed by the NMS contain meaningless regions. If the number of regions is too large, meaningless regions will be input into the model and affect the final performance. The optimal values of hyperparameters in TBAL-Net, such as the reduction ratio and the size of convolutional kernel in attention module, are obtained empirically. The reduction ratio is set to be 16 and the size of convolutional kernel is set to be 3. Under this setting, the number of trainable parameters of TBAL-Net is about 2.7 M, which sits in between MMAL (2.6 M parameters) and NTS-Net (2.8 M parameters). The addition of 0.1 M parameters, compared to MMAL, is mainly from the integration of the attention module, which have provided satisfying return on investment in performance boosting, as demonstrated in Section 4.5.

Similar to [10], there are three components in our CIL method, including cosine normalization (CN), less-forget constraint (LC), and inter-class separation (IS). As shown in the results, (CN+LC+IS) means all these three components are applied to the experiments. There are two different classification strategies used in the experiments, CNN predictions (CNN) and nearest-mean-of-exemplar (NME). Both of these two predictions (Top-1 accuracy) are shown in the results.

In the CIL of CUB-200-2011, 50 classes are randomly selected as the initial training set for training the proposed TBAL-Net and all baselines. In each incremental phase, 10 new classes are fed into the model to train models for recognizing new classes. In the construction process of an exemplar set, 20 samples which are the closest ones to the average prototype of each class are selected. In the CIL of FGVC-Aircraft, half of the total classes (35 classes) are randomly selected as the initial training set for training the proposed TBAL-Net and all baselines. In each incremental phase, five new classes are fed into the model to train models for recognizing new classes of aircraft families. According to the strategy used in the CIL of CUB-200-2011, 20 samples are selected during the construction of an exemplar set. In the Stanford-Car, half of the total classes (98 classes) are randomly selected as the initial training set for training the proposed TBAL-Net and all baselines. In each incremental phase, 14 classes are fed into the model to train models for recognizing new cars. Twenty samples are selected during the construction of an exemplar set. There is no constraint on the total size of exemplar set in our experiments. It is worth noting that in the experiment, the output results of the global branch of TBAL-Net were used as the basis for constructing the exemplar set, in order to avoid feature representation errors caused by localization errors in local regions. Each phase results of three fine-grained datasets are shown in each column of Tables 2–4.

4.4. Ablation Study

To validate the design choices, we evaluated the proposed model under different settings. Specifically, the effects of the attention module and the number of incremental phases are evaluated:

- **Impact of attention module**. Tables 2–4 present the top-1 accuracy of models with and without the attention module. Models with the attention module have a suffix -ATT. It is observed that the addition of the attention mechanism leads to consistent performance improvement for all three datasets in all incremental phases, demonstrating a reliable boosting effect.
- **Impact of incremental phases**. Figures 4–6 show the experimental results of different choices of incremental phases number. For each dataset, we chose two levels, corresponding to a low level and a high level of the incremental phase number, as shown in subfigures (a) and (b), respectively. It is observed that the models perform better with a lower number of incremental phases in all datasets, which is explainable due to the nature of CIL. Essentially, the more incremental phases we have, the less classes per phase, and the more challenges for models to memorize features and patterns learned from previous phases.

Table 2. Performance of TBAL-Net with/without attention modules on CUB-200-2011 (Top-1 Accuracy). The highest score in each column is marked in bold.

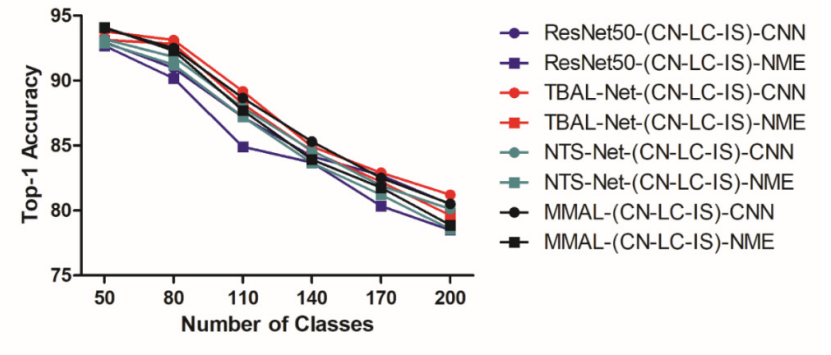
Method	**50**	**60**	**70**	**80**	**90**	**100**	**110**	**120**
TBAL-Net-(CN-LC-IS)-CNN	92.912	91.012	89.793	88.874	87.572	86.476	85.317	84.216
TBAL-Net-(CN-LC-IS)-NME	92.230	90.973	89.741	89.167	87.830	86.376	84.917	83.853
TBAL-Net-(CN-LC-IS)-CNN-ATT	**93.792**	**92.126**	**90.958**	**90.133**	**88.746**	**87.193**	**86.178**	**84.973**
TBAL-Net-(CN-LC-IS)-NME-ATT	93.139	91.772	90.431	89.831	88.103	86.733	85.208	84.187
Method	**130**	**140**	**150**	**160**	**170**	**180**	**190**	**200**
TBAL-Net-(CN-LC-IS)-CNN	82.466	81.831	80.871	80.167	79.301	78.653	78.200	77.467
TBAL-Net-(CN-LC-IS)-NME	81.903	81.013	80.276	79.337	78.498	77.667	76.605	76.031
TBAL-Net-(CN-LC-IS)-CNN-ATT	**83.240**	**82.617**	**81.420**	**80.707**	**79.910**	**79.379**	**79.088**	**78.210**
TBAL-Net-(CN-LC-IS)-NME-ATT	82.740	81.650	80.873	79.921	79.121	78.310	77.141	76.563

Table 3. Performance of TBAL-Net with/without attention modules on FGVC-Aircraft (Top-1 Accuracy).

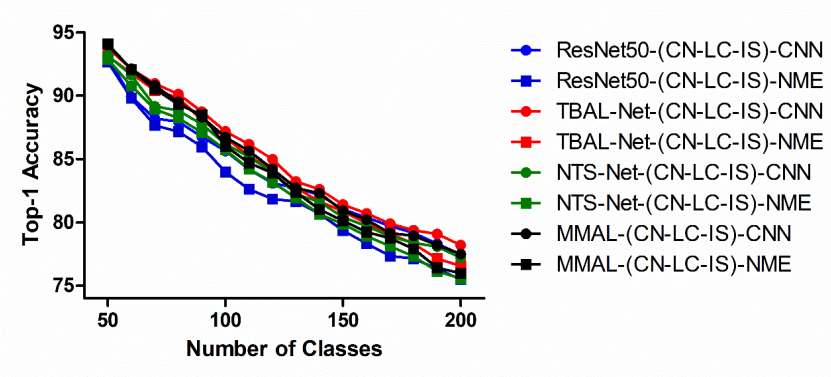
Method	**35**	**40**	**45**	**50**	**55**	**60**	**65**	**70**
TBAL-Net-(CN-LC-IS)-CNN	97.137	96.262	94.137	92.863	92.031	91.073	89.167	88.393
TBAL-Net-(CN-LC-IS)-NME	97.330	95.167	94.033	89.167	88.030	86.737	85.317	83.973
TBAL-Net-(CN-LC-IS)-CNN-ATT	97.846	**96.91**	94.87	**93.65**	**92.879**	**91.798**	**89.86**	**89.08**
TBAL-Net-(CN-LC-IS)-NME-ATT	**98.012**	96.12	**94.96**	93.45	92.325	91.02	89.658	88.233

Table 4. Performance of TBAL-Net with/without attention modules on Stanford-Car (Top-1 Accuracy).

Method	**98**	**112**	**126**	**140**	**154**	**168**	**182**	**196**
TBAL-Net-(CN-LC-IS)-CNN	95.863	93.317	92.073	89.915	88.767	87.876	86.717	85.916
TBAL-Net-(CN-LC-IS)-NME	95.315	93.073	91.930	89.390	88.130	87.176	86.527	85.353
TBAL-Net-(CN-LC-IS)-CNN-ATT	**96.74**	**94.215**	**93.013**	**90.87**	**89.706**	**88.942**	**88.021**	**87.312**
TBAL-Net-(CN-LC-IS)-NME-ATT	96.317	93.915	92.813	89.77	88.916	88.342	87.821	86.93

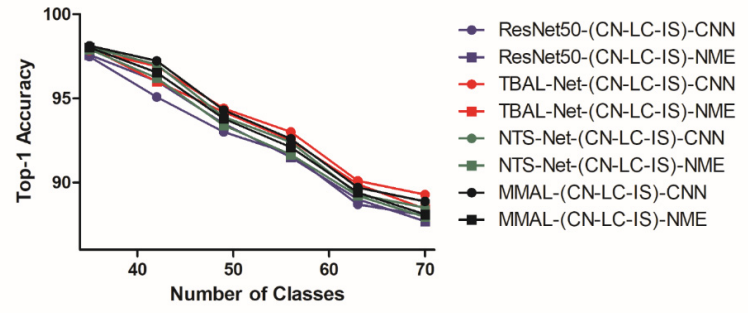


(**a**) The number of incremental phases is 5.

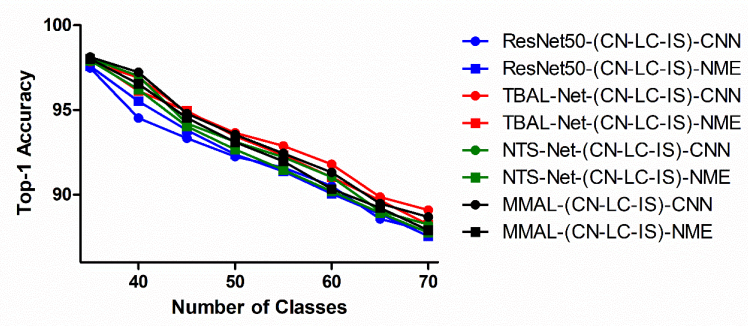


(**b**) The number of incremental phases is 15.

Figure 4. The performance of different number of incremental phases on CUB-200-2011.

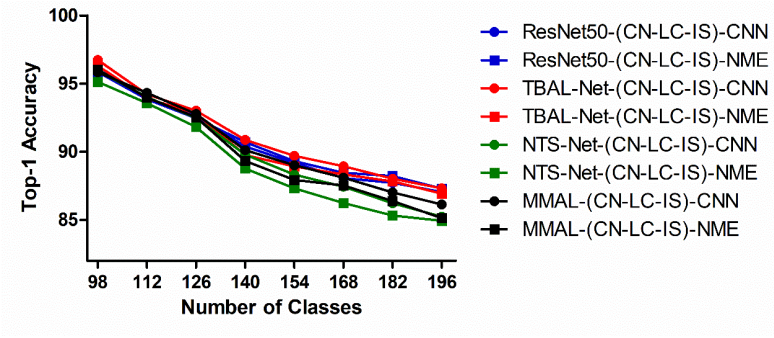


(**a**) The number of incremental phases is 5.

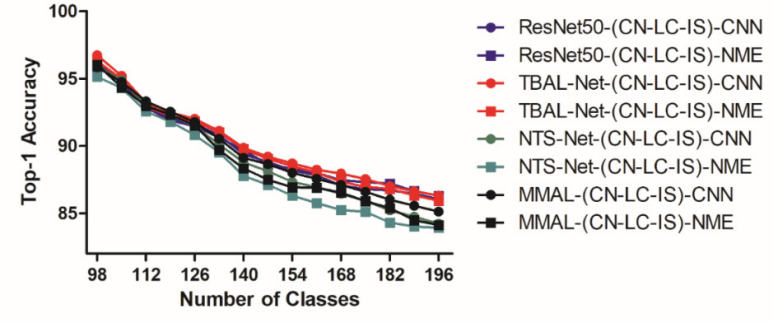


(**b**) The number of incremental phases is 7.

Figure 5. The performance of different number of incremental phases on FGVC-Aircraft.



(**a**) The number of incremental phases is 7.



(**b**) The number of incremental phases is 14.

Figure 6. The performance of different number of incremental phases on Stanford-Car.

4.5. Results

Evaluation on CUB-200-2011. Tables 5 and 6 and Figure 4 show the experimental results on CUB-200-2011. Both CNN and NME predictions of the proposed TBAL-Net outperform the baselines. It is observed that MMAL presents better performance in the initial two phases than other methods, showing its ability to capture more distinguishable patterns in the beginning of the CIL training when the number of classes is relatively low. However, as more new classes participate into training, the proposed TBAL-Net starts to outperform its peers. It is shown that as the number of classes went beyond 70, the TBAL-Net-(CN-LC-IS)-CNN model consistently outperforms other methods. Furthermore, TBAL-Net with the CNN prediction is better than TBAL-Net with the NME prediction, showing that the former demonstrates a superior ability in extracting more discriminative fine-grained features. Moreover, it is noted that the localization modules designed for FGVC, such as region proposal network (RPN) in NTS-Net, may be not suitable for fine-grained CIL. The potential reasons are: (1) the RPN is randomly initialized, and (2) due to the limited data size per category, RPN may not be trained well.

Table 5. Performance (Top-1 Accuracy %) on CUB-200-2011 as the number of classes increases from 50 to 120.

Method	50	60	70	80	90	100	110	120
ResNet50-(CN-LC-IS)-CNN	92.968	89.924	88.200	87.957	86.749	85.650	84.187	83.088
ResNet50-(CN-LC-IS)-NME	92.686	89.807	87.650	87.174	85.978	83.986	82.616	81.847
NTS-Net-(CN-LC-IS)-CNN	93.192	91.626	89.158	88.833	87.746	86.193	85.178	83.973
NTS-Net-(CN-LC-IS)-NME	92.891	90.772	88.930	88.231	87.103	85.733	84.208	83.187
MMAL-(CN-LC-IS)-CNN	94.018	**92.130**	90.810	89.502	88.310	86.730	85.653	84.210
MMAL-(CN-LC-IS)-NME	**94.107**	92.070	90.531	89.312	88.512	86.037	84.702	83.903
TBAL-Net-(CN-LC-IS)-CNN	93.792	92.126	**90.958**	**90.133**	**88.746**	**87.193**	**86.178**	**84.973**
TBAL-Net-(CN-LC-IS)-NME	93.139	91.772	90.431	89.831	88.103	86.733	85.208	84.187

Table 6. Performance (Top-1 Accuracy %) on CUB-200-2011 as the number of classes increases from 130 to 200.

Method	130	140	150	160	170	180	190	200
ResNet50-(CN-LC-IS)-CNN	82.716	82.176	81.032	80.390	79.707	79.179	78.319	77.477
ResNet50-(CN-LC-IS)-NME	81.651	80.692	79.366	78.332	77.346	77.145	76.337	75.492
NTS-Net-(CN-LC-IS)-CNN	82.240	81.617	80.420	79.707	78.910	78.379	78.088	77.210
NTS-Net-(CN-LC-IS)-NME	81.940	80.650	79.873	78.921	78.121	77.310	76.141	75.563
MMAL-(CN-LC-IS)-CNN	82.720	82.312	80.921	80.218	79.150	78.940	78.210	77.501
MMAL-(CN-LC-IS)-NME	82.390	81.030	80.127	79.238	78.750	77.913	76.420	75.980
TBAL-Net-(CN-LC-IS)-CNN	**83.240**	**82.617**	**81.420**	**80.707**	**79.910**	**79.379**	**79.088**	**78.210**
TBAL-Net-(CN-LC-IS)-NME	82.740	81.650	80.873	79.921	79.121	78.310	77.141	76.563

Evaluation on FGVC-Aircraft. Table 7 and Figure 5b show the experimental results on FGVC-Aircraft. We can observe similar results as CUB-200-2011. In the initial two phases, MMAL has demonstrated better performance, and TBAL-Net catches up since the third phase when the number of classes reaches 45. It is also observed that the performance differences among the compared models are relatively small. This may be caused by the lower number of classes of the FGVC-Aircraft dataset, which leads to more samples per class, allowing each model to learn more informative features.

Table 7. Performance (Top-1 Accuracy %) on FGVC-Aircraft as the number of classes increases from 35 to 70.

Method	35	40	45	50	55	60	65	70
ResNet50-(CN-LC-IS)-CNN	97.48	94.519	93.33	92.248	91.595	90.506	88.566	87.849
ResNet50-(CN-LC-IS)-NME	97.587	95.52	93.816	92.37	91.37	90.052	88.85	87.549
NTS-Net-(CN-LC-IS)-CNN	98.04	97.012	94.21	93.12	92.21	91.031	89.012	88.233
NTS-Net-(CN-LC-IS)-NME	97.921	96.21	94.037	92.67	91.47	90.19	88.921	87.75
MMAL-(CN-LC-IS)-CNN	**98.127**	**97.23**	94.796	93.545	92.439	91.32	89.47	88.676
MMAL-(CN-LC-IS)-NME	98.021	96.543	94.539	93.098	91.97	90.33	89.215	87.901
TBAL-Net-(CN-LC-IS)-CNN	97.846	96.91	94.87	**93.65**	**92.879**	**91.798**	**89.86**	**89.08**
TBAL-Net-(CN-LC-IS)-NME	98.012	96.12	**94.96**	93.45	92.325	91.02	89.658	88.233

Evaluations on Stanford-Car. Table 8 and Figure 6a show the experimental results on Stanford-Car. Similar to the previous two datasets, the only competition of TBAL-Net is MMAL, which presents a better accuracy than TBAL-Net at phase 2. However, TBAL-Net dominates MMAL and other models in all of the other incremental phases.

Table 8. Performance (Top-1 Accuracy %) on Stanford-Car as the number of classes increases from 98 to 196.

Method	98	112	126	140	154	168	182	196
ResNet50-(CN-LC-IS)-CNN	96.053	93.965	92.620	90.392	89.182	88.069	87.716	87.054
ResNet50-(CN-LC-IS)-NME	95.854	93.857	92.466	90.704	89.308	88.460	**88.212**	87.303
NTS-Net-(CN-LC-IS)-CNN	95.940	94.021	92.473	89.794	88.329	87.440	86.217	85.233
NTS-Net-(CN-LC-IS)-NME	95.137	93.566	91.815	88.779	87.316	86.242	85.321	84.930
MMAL-(CN-LC-IS)-CNN	95.83	**94.32**	92.778	90.121	89.012	88.103	87.02	86.133
MMAL-(CN-LC-IS)-NME	96.03	93.97	92.531	89.33	87.93	87.531	86.39	85.127
TBAL-Net-(CN-LC-IS)-CNN	**96.74**	94.215	**93.013**	**90.87**	**89.706**	**88.942**	88.021	**87.312**
TBAL-Net-(CN-LC-IS)-NME	96.317	93.915	92.813	89.77	88.916	88.342	87.821	86.93

5. Conclusions

In this paper, we propose TBAL-Net, which contains an attention module similar to [15] and a localization module for fine-gained CIL. We adopt the CIL framework introduced in [10] and evaluate ResNet50, NTS-Net, MMAL, and the proposed TBAL-Net on three fine-grained object datasets, including CUB-200-2011 [18], FGVC-Aircraft [19], and Stanford Cars [20]. As an effective FGVC method, MMAL can achieve better performance than other methods in the initial phase but has lower performance than the proposed TBAL-Net. NTS-Net can also achieve good performance in the initial phase, but its performance is lower than other methods on all three fine-grained object datasets. As a traditional and effective network architecture, ResNet50 outperforms both NTS-Net and MMAL on Stanford-Car in CIL. These results lead to the following conclusions:

(1) The localization modules designed for FGVC, such as region proposal network (RPN) in NTS-Net, may be not suitable for fine-grained CIL. The RPN is randomly initialized. Due to the limited data size, RPN may not be trained well.
(2) The localization modules in MMAL only increases few parameters. Additionally, MMAL can achieve good performance on FGVC. MMAL does not have enough learning ability in fine-grained CIL.
(3) The attention module similar to [36] is effective in fine-grained CIL. Therefore, TBAL-Net can extract a lot of discriminative fine-grained features in the experiments, as shown in the NME predictions.

Author Contributions: Conceptualization, J.G. and X.L.; methodology, G.Q.; software, J.G.; validation, G.Q., S.X. and X.L.; writing—original draft preparation, J.G.; writing—review and editing, X.L. and G.Q.; supervision, S.X. and X.L.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Ministry of Education 2021 University-Industry Cooperation Project, China (202002018063, 9 November 2020–9 November 2021), under the project entitled "Virtual prototype-based autonomous driving of miniature intelligent vehicles." The funding agency has no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Data Availability Statement: The CUB-200-2011 [9] (http://www.vision.caltech.edu/visipedia/CUB-200-2011.html (accessed on 2 June 2021)), FGVC-Aircraft [10] (https://www.robots.ox.ac.uk/~vgg/data/fgvc-aircraft/ (accessed on 2 June 2021)) and Stanford-Cars [11] (https://ai.stanford.edu/~jkrause/cars/car_dataset.html (accessed on 2 June 2021)) data sets presented in this work are publicly available.

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

References

1. Dang, S.; Cao, Z.; Cui, Z.; Pi, Y.; Liu, N. Class Boundary Exemplar Selection Based Incremental Learning for Automatic Target Recognition. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5782–5792. [CrossRef]
2. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
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.
4. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
5. 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.
6. Shmelkov, K.; Schmid, C.; Alahari, K. Incremental Learning of Object Detectors without Catastrophic Forgetting. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
7. Zhang, J.; Zhang, J.; Ghosh, S.; Li, D.; Tasci, S.; Heck, L.; Zhang, H.; Kuo, C.-C.J. Class-incremental Learning via Deep Model Consolidation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 1–5 March 2020.
8. Masana, M.; Liu, X.; Twardowski, B.; Menta, M.; Bagdanov, A.D.; van de Weijer, J. Class-incremental learning: Survey and performance evaluation on image classification. *arXiv* **2020**, arXiv:2010.15277.

9. Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. iCaRL: Incremental Classifier and Representation Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
10. Hou, S.; Pan, X.; Loy, C.C.; Wang, Z.; Lin, D. Learning a Unified Classifier Incrementally via Rebalancing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
11. Castro, F.M.; Marín-Jiménez, M.J.; Guil, N.; Schmid, C.; Alahari, K. *End-to-End Incremental Learning*; Springer: Singapore, 2018.
12. Liu, Y.; Schiele, B.; Sun, Q. Meta-Aggregating Networks for Class-Incremental Learning. *arXiv* **2020**, arXiv:2010.05063.
13. Li, Z.; Hoiem, D. Learning without Forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2935–2947. [CrossRef] [PubMed]
14. Mermillod, M.; Bugaiska, A.; Bonin, P. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Front. Psychol.* **2013**, *4*, 504. [CrossRef] [PubMed]
15. Krizhevsky, A.; Hinton, G. Learning multiple layers of features from tiny images. In *Handbook of Systemic Autoimmune Diseases*; Elsevier: Amsterdam, The Netherlands, 2009; Volume 1, p. 4.
16. Russakovsky, O.; Deng, J.; Karpathy, A.; Ma, S.; Russakovsky, O.; Huang, Z.; Bernstein, M.; Krause, J.; Su, H.; Fei-Fei, L.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
17. Liu, Y.; Schiele, B.; Sun, Q. Adaptive Aggregation Networks for Class-Incremental Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
18. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. The Caltech-UCSD Birds200-2011 Dataset. *Adv. Water Resour.* 2011. Available online: https://authors.library.caltech.edu/27452/ (accessed on 20 November 2021).
19. Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; Vedaldi, A. Fine-Grained Visual Classification of Aircraft. *arXiv* **2013**, arXiv:1306.5151.
20. Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. 3D Object Representations for Fine-Grained Categorization. In Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops, Sydney, NSW, Australia, 2–8 December 2013.
21. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef] [PubMed]
22. Kemker, R.; Abitino, A.; McClure, M.; Kanan, C. Measuring Catastrophic Forgetting in Neural Networks. *arXiv* **2018**, arXiv:1708.02072.
23. Yuan, L.; Tay, F.E.; Li, G.; Wang, T.; Feng, J. Revisiting Knowledge Distillation via Label Smoothing Regularization. *arXiv* **2020**, arXiv:1909.11723.
24. Shi, Y.; Hwang, M.-Y.; Lei, X.; Sheng, H. Knowledge Distillation for Recurrent Neural Network Language Modeling with Trust Regularization. *arXiv* **2019**, arXiv:1904.04163.
25. Yun, S.; Park, J.; Lee, K.; Shin, J. Regularizing Class-Wise Predictions via Self-Knowledge Distillation. *arXiv* **2020**, arXiv:2003.13964.
26. Yuan, L.; Tay, F.E.H.; Li, G.; Wang, T.; Feng, J. Revisit Knowledge Distillation: A Teacher-free Framework. *arXiv* **2019**, arXiv:1909.11723.
27. Kim, K.; Ji, B.; Yoon, D.; Hwang, S. Self-Knowledge Distillation with Progressive Refinement of Targets. *arXiv* **2020**, arXiv:2006.12000.
28. Wang, Y.; Li, H.; Chau, L.-P.; Kot, A.C. Embracing the Dark Knowledge: Domain Generalization Using Regularized Knowledge Distillation. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, 20–24 October 2021.
29. Zhao, L.; Peng, X.; Chen, Y.; Kapadia, M.; Metaxas, D.N. Knowledge as Priors: Cross-Modal Knowledge Generalization for Datasets Without Superior Knowledge. *arXiv* **2020**, arXiv:2004.00176.
30. Liu, C.; Xie, H.; Zha, Z.-J.; Ma, L.; Yu, L.; Zhang, Y. Filtration and Distillation: Enhancing Region Attention for Fine-Grained Visual Categorization. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11555–11562.
31. Kermany, D.S.; Goldbaum, M.; Cai, W.; Valentim, C.C.; Liang, H.; Baxter, S.; McKeown, A.; Yang, G.; Wu, X.; Yan, F.; et al. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell* **2018**, *172*, 1122–1131.e9. [CrossRef] [PubMed]
32. Niu, Y.; Jiao, Y.; Shi, G. Attention-shift based deep neural network for fine-grained visual categorization. *Pattern Recognit.* **2021**, *116*, 107947. [CrossRef]
33. Yang, Z.; Luo, T.; Wang, D.; Hu, Z.; Gao, J.; Wang, L. *Learning to Navigate for Fine-Grained Classification*; Springer: Berlin, Germany, 2018.
34. Zhang, F.; Li, M.; Zhai, G.; Liu, Y. Multi-branch and Multi-scale Attention Learning for Fine-Grained Visual Categorization. *arXiv* **2020**, arXiv:2003.09150.
35. Fu, J.; Zheng, H.; Mei, T. Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
36. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. *CBAM: Convolutional Block Attention Module*; Springer: Singapore, 2018.