Deep Transfer Learning Method using Self-Pixel and Global Channel Attentive Regularization

Changhee Kang and Sang-ug Kang *

Posted Date: 22 April 2024

doi: 10.20944/preprints202404.1423.v1

Keywords: deep transfer learning; knowledge distillation; regularization

 doi:10.20944/preprints202404.1423.v1

Article

Deep Transfer Learning Method Using Self-Pixel and Global Channel Attentive Regularization

Changhee Kang and Sang-ug Kang *

Department of Computer Science, Sangmyung University; 202032028@sangmyung.kr
* Correspondence: sukang@smu.ac.kr; Tel.: +82-2-781-7588

Abstract: The purpose of this paper is to propose a novel transfer learning regularization method based on knowledge distillation. Recently, transfer learning methods have been used in various fields. However, the problems such as knowledge loss still occur during the process of transfer learning to a new target dataset. To solve these problems, there are various regularization methods based on knowledge distillation techniques. In this paper, we propose a transfer learning regularization method based on feature map alignment used in the field of knowledge distillation. The proposed method is composed of two attention based submodules: self-pixel attention (SPA) and global channel attention (GCA). The self-pixel attention submodule utilizes both the feature maps of the source and target models, so that it provides an opportunity to jointly consider the features of the target and the knowledge of the source. The global channel attention submodule determines the importance of channels through all layers, unlike the existing methods that calculate these only within a single layer. Accordingly, transfer learning regularization is performed by considering both the internal of each single layer and the depth of the entire layer. Consequently, the proposed method using both of these submodules showed overall improved classification accuracy than the existing methods in classification experiments on commonly used dataset.

Keywords: deep transfer learning; knowledge distillation; regularization

1. Introduction

In recent computer vision literature, deep learning approaches are being used in various fields. Hussein et al. [1] applied a deep network to the medical field to classify lung and pancreatic tumor. Ramesh et al. [2] devised a text-to-image generator that interprets the meaning of input text and then creates an image containing the interpreted meaning. Also, Feng et al. [3] proposed an object segmentation method that divides objects such as pedestrians, vehicles, roads, and traffic lights for autonomous driving. Kang et al. [4] replaced traditional denoising image filters with a single recursive neural network to remove various types of unwanted signals. For most of these tasks, supervised learning performed better when the amount of training dataset was sufficient as demonstrated by Orabona et al. in [5] with the well-known MNIST dataset. However, since data collection and labeling are time consuming and costly, transfer learning approaches have emerged.

Transfer learning aims to perform new target tasks on small-scale data by leveraging deep neural network knowledge pre-trained on large-scale data as shown in [6] and [7]. For example, various deep models like ResNet [9], VGG [10] were pre-trained on large-scale datasets such as ImageNet [8] ,and then utilized for new target tasks. Using a pre-trained model, also called the source model, as a starting point, you can use fine-tuning techniques to transform it into a newly trained model for a new target task. Some of the weights of the source model are changed during separate training sessions using a new target dataset in order to create new target models for new tasks [11–14]. In general, the performance of fine-tuning techniques, such as convergence speed and prediction accuracy, is better than traditional supervised learning. For example, Ng et al. [15] showed about 16% higher emotion recognition accuracy, Zhao [16] achieved 7.2% better classification results, and Mohammadian et al. [17] attained about 12.1% improvement over the conventional approach [18] in diabetes diagnosis. However, there are two problems with the fine-tuning approaches. First, the distribution of source

 doi:10.20944/preprints202404.1423.v1

and target datasets should be similar. Wang et al. [19] defined negative transfer learning as the phenomenon in which source knowledge interferes with target learning when the source and target domains are not sufficiently similar. It was demonstrated by showing that the larger the gap between the source and target domains, the lower the transfer learning performance. Second, fine-tuning the target model often loses some useful knowledge for the target task learned from the source model, even when there is only a slight distributional difference between the source and target datasets, as demonstrated in [20].

To cope with these problems, the L2 regularization is applied to the weights of target model during the fine-tuning process. Li et al. [21] proposed the $L^2 - SP$ method, which encourages the weights of the fine-tuned target model to be similar to those of the source model, called the starting point (SP). This $L^2 - SP$ regularization method showed about 0.8% to 8.6% better results than the vanilla L2 regularization method [21,22]. However, weight regularization approaches often fail to converge the target model or often lose useful knowledge learned from the source model, as it is difficult to find the appropriate regularization strength due to optimization sensitivity [20]. To address this problem, Hinton et al. [23] proposed the knowledge distillation method ,which extracts only the necessary knowledge from the source model, rather than all of it, and transfers it to the target model. It also allows for different source and target model structures, typically large for the source and simple for the target. Therefore, the target model is trained utilizing feature maps of the source model, instead of weights, as in [24]. Therefore, both source model and target model are necessary during the training session of the target model. Utilizing the entire spatial area and all channels of the feature map is sometimes not effective, so methods have been proposed to use only the parts that are actually influential [26,27]. Mirzadeh et al. [26] proposed a distillation method using a teacher assistant model, which is an intermediate size between the teacher and student models. Li et al. [27] added a 1×1 convolution layer to each specific layer of the student model to make its feature map similar to the corresponding feature map of the teacher model. Li and Hoiem [28] proposed a transfer learning method called LWF (Learning Without Forgetting), which can learn a new task while retaining the knowledge and capabilities originally learned. The LWF has integrated the knowledge distillation method into the transfer learning process so that it is possible to learn without forgetting the original knowledge, even when using only dataset for a new task. Li et al. [29] proposed the DELTA (Deep Learning Transfer using Feature Map with Attention) method, which assigns attention scores to feature maps based on the LWF method.

The DELTA method [29] determines the importance of each filter by calculating the loss value using a feature extractor model($L^2 - FE$) that determines the useful of the filter. $L^2 - FE$ is trained only the fully-connected layers of the source model using the target data. After filling each filter of a specific layer from the $L^2 - FE$ model with a value of 0, the importance of the filters is calculated according to the changing loss value between prediction and label. Through the calculated importance of source model filters, the target model trained with a regularization method using an attention mechanism [30,31] that gives weights to filters containing useful knowledge in the source model. Xie et al. [32] proposed Attentive Feature Alignment (AFA) based on a knowledge distillation paradigm [24] similar to DELTA. AFA extracts attention weights in the spatial and channel information related to the target from the feature map extracted from the source model through the additional submodule networks. While DELTA calculates the importance of a convolution filter using the subtraction in loss values, AFA calculates the importance of a convolution filter using a submodule network defined as an attentive module. The attentive module consists of two types of modules that receive the feature map extracted from the convolutional layer and calculate the importance of the convolutional filter by reflecting spatial or channel information. Compared to DELTA method, AFA considers attention to the spatial information as well as channels and uses a method of calculating weights through a submodule network.

Both the AFA [32] and DELTA [29] methods determine the relative importance of the source model filters in the target models and represent it as real numbers ranging from 0 to 1, all summing to

 doi:10.20944/preprints202404.1423.v1

1. However, the importance comparison is evaluated only within the scope of a single convolution layer, i.e. the same value in different convolution layers has the same importance throughout the target model. Since different convolution layers have different roles, e.g. simple functionalities for input side layers and vice versa, it is natural that each convolution layer has a different impact on the target model. Therefore, the relative importance of a filter should also be determined considering the position of a convolution layer. In this paper, the importance of filters is not compared within a single convolution layer, but across all layers. Thus, we propose a global channel attention module based on the SENet method [33]. In addition, we propose a self-pixel attention module that regularizes with the feature map of the target model as well as the feature map of the source model, which extends the concept of spatial attention module proposed in AFA. The proposed regularization method using the two proposed attention modules shows improved classification accuracy and convergence speed compared to existing regularization methods. The contents of this paper are organized as follows. Section 2 explains related works, Section 3 describes the proposed regularization model for transfer learning, Section 4 explains the experimental settings such as the dataset and hyperparameters used in the experiment, and Section 5 describes the experimental results. Finally, the last Section 6 concludes the paper.

2. Related Works

The AFA method [32] uses two submodules, AST (Attentive Spatial Transfer) and ACT (Attentive Channel Transfer), which take the feature map of the source model as input and calculate the attention values for regularized optimization of the target model. The AST module calculates feature-specific attention values and the ACT module outputs channel-specific values. The AST and ACT calculate weighting values for spatial positions and channels of feature maps respectively. Since feature maps are used to regularize the optimization, we need to define the feature map first as shown in Equation (1).

$$FM^i_{S\,or\,T} = f(x, W^i_{S\,or\,T}) \tag{1}$$

where the superscript i and the subscripts S, T refer to the i'th convolutional layer of the source model or the target model used for regularization, and $W^i_{S\,or\,T}$ and x denote the weights of i'th layer and an input image, respectively. The feature map of the AST network can be derived by flattening the FM^i_S along the height and width directions using the flatten function $Flat(\cdot) : \mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{C\times(HW)}$, and then average pooling the result along the channels using the function $AvgPool(\cdot) : \mathbb{R}^{C\times(HW)} \rightarrow \mathbb{R}^{1\times(HW)}$. The attention weights AST^i can be calculated using Equation (2) and then the AST loss L_{AST} is calculated using Equation (3), respectively.

$$AST^i = Softmax(FC^i_{AST}(AvgPool(Flat(FM^i_S)))) \tag{2}$$

$$L_{AST} = \sum_{i=1}^{n} \frac{1}{2} \| AST^i \cdot (FM^i_S - FM^i_T) \|^2_F \tag{3}$$

where the $FC_{AST}(\cdot) : \mathbb{R}^{1\times(HW)} \rightarrow \mathbb{R}^{HW}$ consists of two fully-connected layers, and n denotes the total number of convolutional layers selected to extract feature maps. The difference between FM^i_S and FM^i_T is multiplied by the attention weights element by element, and then the Frobenius norm of the vector is computed in order to find the AST loss of the i'th convolutional layer. The AST loss L_{AST} is the sum of all the losses of the n selected convolutional layers. By minimizing L_{AST}, the original knowledge is differentially transferred to the target model. The ACT module weighs the importance of the channel information in the source feature maps. The attention weights ACT^i can be calculated using Equation (4).

$$ACT^i = Softmax(FC^i_{ACT}(Flat(FM^i_S))) \tag{4}$$

 doi:10.20944/preprints202404.1423.v1

Unlike the AST submodule, the ACT submodule only applies a flattening function and no average pooling to obtain the transformed feature map [32]. Therefore, the feature map is transformed using the flatten function $Flat(\cdot) : \mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{C\times(HW)}$, and finally, ACT^i is determined by $FC_{ACT}(\cdot) : \mathbb{R}^{C\times(HW)} \rightarrow \mathbb{R}^{C}$. The ACT loss L_{ACT} is expressed by Equation (5).

$$L_{ACT} = \sum_{i=1}^{n} \frac{1}{2} ACT^i \cdot \|(FM_S^i - FM_T^i)\|_F^2 \tag{5}$$

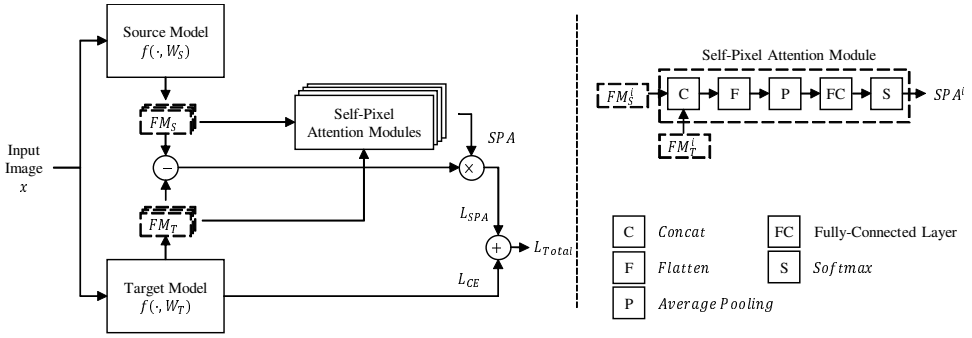
After calculating the difference between two feature maps, the attention weight values calculated by the ACT module are multiplied by the magnitudes of the channel vectors. The L_{ACT} is obtained by adding all the loss values of the selected n convolutional layers. For the whole training period, we devote half of the epochs to AST and the other half to ACT. The transfer learning objective function is as follows.

$$L_{Total} = L_{CE} + \alpha \cdot L_{AST\,or\,ACT} + \beta \cdot WD \tag{6}$$

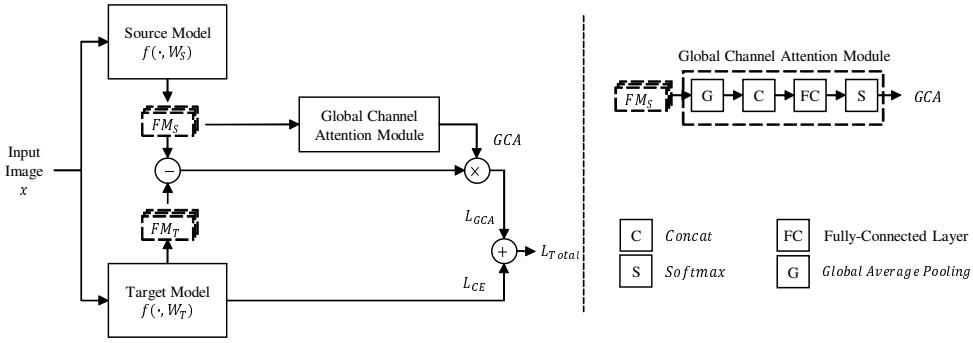
where L_{CE} is cross-entropy loss function and WD is weight decay. L2 regularization is applied to the weights of the fully-connected layers of the target model. The values of α and β are the coefficients used for each loss value.

3. The Proposed Method

The two submodules SPA (Self-Pixel Attention) and GCA (Global Channel Attention) are proposed as shown in Figure 1. The SPA extends the AST in [32] in order to strengthen the related knowledge between the source and the target models by utilizing both their feature maps. The GCA calculates the channel importance of all convolutional layers together, rather than per layer as in [32].



(a) Overview of the training process with SPA (Self-Pixel Attention) module



(b) Overview of the training process with GCA (Global Channel Attention) module

Figure 1. Block diagram of the proposed regularization method: (**a**) SPA module, (**b**) GCA module

 doi:10.20944/preprints202404.1423.v1

3.1. The Self-Pixel Attention Submodule

AFA [32] applied pixel attention to consider the importance of spatial information, and showed improved performance than DELTA [29], which did not consider spatial information. However, the previous pixel attentive module simply utilized only the spatial information of the feature map extracted from the source model to calculate the attention weights. The proposed SPA calculates attention weights by concatenating FM_S^i and FM_T^i to exploit the spatial information of the source and target models in the regularization process of training.

$$SPA^i = Softmax(FC_{SPA}^i(AvgPool(Flat(Concat(FM_S^i, FM_T^i))))) \tag{7}$$

The concatenation is performed along the channel direction using the function $Concat(\cdot) : \mathbb{R}^{C_S \times H \times W}, \mathbb{R}^{C_T \times H \times W} \rightarrow \mathbb{R}^{(C_S+C_T) \times H \times W}$. Flatten and average pooling functions transform the dimension of the feature map to $\mathbb{R}^{1 \times (HW)}$. The fully-connected layer $FC_{SPA}^i(\cdot)$ followed by the softmax function outputs the attention weights of the i'th convolutional layer. Finally, the SPA loss is calculated as in Equation (8), similar to Equation (3).

$$L_{SPA} = \sum_{i=1}^{n} \frac{1}{2} \| SPA^i (FM_S^i - FM_T^i) \|_F^2 \tag{8}$$

3.2. The Global Channel Attention Submodule

Equal attention weight means that the corresponding channels have the same importance in the previous works, regardless of the importance of the convolutional layer that includes them. For example, the deeper layers usually contain more information or knowledge than the shallower layers. Therefore, the proposed GCA calculates the relative importance of channels in all selected convolutional layers.

$$GCA = Softmax(FC_{GCA}(Concat(GAP(FM_S^1), \cdots, GAP(FM_S^n)))) \tag{9}$$

The feature map of source model FM_S^i is the input to the global average pooling function $GAP(\cdot) : \mathbb{R}^{C \times H \times W} \rightarrow \mathbb{R}^{C \times 1}$. The final GCA vector is in the space of $\mathbb{R}^{C_1+C_2+\cdots+C_n}$. The GCA loss is computed as in Equation (10), similar to Equation (5).

$$L_{GCA} = \sum_{i=1}^{n} \frac{1}{2} GCA^i \| (FM_S^i - FM_T^i) \|_F^2 \tag{10}$$

The subtraction between the feature maps of the i'th convolutional layer obtained from the source and target models is multiplied for each channel by the GCA obtained through Equation (9). GCA^i is a channel attention weight corresponding to the i'th convolutional layer and has the size of $\mathbb{R}^{C_i}$, which is a part of GCA. By applying this GCA loss, the target model is regularized in the channel direction, and in the source model, not only the channel attention of a single layer but also the relative emphasis according to the depth is progressed. The structure of the fully-connected layers used in the proposed method can be confirmed by Table 2.

3.3. Objective Function for Regularization

The optimization process is divided into two stages and is similar to the previous work, AFA [32]. In the first stage, we use a pre-trained model such as ImageNet and Places 365, selected according to

 doi:10.20944/preprints202404.1423.v1

the nature of the target task, as the source model. The weights of the convolutional layer of the source model are transferred to the target model, and those of the SPA submodule and the fully connected layers of the target model are randomly initialized. The target model is then trained during the first half epochs using the loss function in Equation (11) derived from Equation (6).

$$L_{Total} = L_{CE}(f(x, W_T), y) + \alpha \cdot L_{SPA} + \beta \cdot WD \tag{11}$$

The L_{CE} is determined by calculating the cross entropy of the predicted and ground distributions. The WD is the L2 regularization weight decay applied to the weights of the fully connected layers of the target model. In the second stage, the weights from the target model trained in the first stage are sent back to the source model because they contain better knowledge than the original weights from the source. Then, only the weights of the GCA submodule are initialized randomly, and the loss function in Equation (12) is used to train the target model for the remaining epochs.

$$L_{Total} = L_{CE}(f(x, W_T), y) + \alpha \cdot L_{GCA} + \beta \cdot WD \tag{12}$$

4. Details of the Experiments

4.1. Dataset Setup

We evaluate the performance of the proposed method through object and scene classification. For object classification, the source model was pre-trained using ImageNet [8], and the target model used Stanford Dogs 120[35], Caltech 256-30 [36], Caltech 256-60 [36], and CUB-200-2011 [37] datasets. For scene classification, the source model was pre-trained utilizing the Places 365 dataset [34], and the target model was trained and tested using the MIT Indoor 67 dataset [38]. The purpose and characteristics of each target dataset are summarized in Table 1. The Stanford Dogs 120 dataset contains puppies by breed, and consists of a total of 20,580 images for 120 breeds. Caltech 256 is an object recognition dataset containing 30,607 real-world images, of different sizes, spanning 257 object classes, consisting of 256 object classes and an additional clutter class. Each class is represented by at least 80 images. Caltech 256-30 randomly selects 50 images from the Caltech 256 dataset and divides them into 30 and 20 training and test images respectively. Similarly, Caltech 256-60 selects 80 images and divides them into 60 and 20. The CUB-200-2011 dataset is a fine-grained classification of birds by breed, with a total of 200 breeds and 11,788 images. MIT Indoor 67 is a dataset of 67 indoor scenes with a total of 15,620 images. However, only 6,700 of them are used in this experiment: 5,360 for training and 1,340 for testing. During the experiment, the images are resized to 256x256 and then cropped to 224x224 at random locations to use as input to the model. The data configuration and image pre-processing methods were set to be as similar as possible to existing studies.

Table 1. The purpose and characteristics of target datasets

Target dataset	Task	Train samples	Test samples	Classes
Stanford Dogs 120	Object Classification	12,000	8,580	120
Caltech 256-30	Object Classification	7,710	5,140	257
Caltech 256-60	Object Classification	15,420	5,140	257
CUB-200-2011	Object Classification	5,994	5,794	200
MIT Indoor 67	Scene Classification	5,360	1,340	67

4.2. The Structure of Network and Hyperparameters

Most experiments are performed using the ResNet-101 model [9] pre-trained with ImageNet. However, for our experiments on the MIT Indoor 67 dataset, we used the ResNet-50 model because it is the only pre-trained model available for the Places 365 dataset. For model optimization, the SGD is

 doi:10.20944/preprints202404.1423.v1

used with the momentum set to 0.9 and the batch size set to 64. SPA and GCA are trained sequentially for 4,500 iterations each out of a total of 9,000 training iterations. The initial learning rate is 0.01 and decreases to 1/10 at two points: two-thirds of the way through the SPA training period and two-thirds of the way through the GCA training period, eventually becoming 0.0001. In this experiment, the learning rate decay occurs at 3,000 and 7,500 iterations. In addition, r of the fully-connected layer used in the GCA submodule is set to 4. The weighting factor α, which means the strength of the loss value calculated from the submodules, is set in the range of 0.005 to 0.1 depending on the target dataset. The weighting factor β, the intensity of the L2 weight decay, is set to 0.01. The feature maps used as input to the submodules are extracted from a total of four intermediate layers of the source and target models, and are chosen to be the same as in DELTA [29] and AFA [32] for a fair comparison. The selected intermediate layers can be found in Table 3.

Table 2. The structure of submodule networks

Model name	Layer type	Parameter	Value
FC_{SPA}	Fully connected	Input size	$\mathbb{R}^{H\times W}$
		Output size	$\mathbb{R}^{H}$
		Activation	ReLU
	Fully connected	Input size	$\mathbb{R}^{H}$
		Output size	$\mathbb{R}^{H\times W}$
FC_{GCA}	Fully connected	Input size	$\mathbb{R}^{C_1+\cdots+C_n}$
		Output size	$\mathbb{R}^{(C_1+\cdots+C_n)/r}$
		Activation	ReLU
	Fully connected	Input size	$\mathbb{R}^{(C_1+\cdots+C_n)/r}$
		Output size	$\mathbb{R}^{C_1+\cdots+C_n}$

Table 3. The selected intermediate layers in the experiments

Layer index	ResNet-101	ResNet-50
1	Resnet.layer1.2.conv3	Resnet.layer1.2.conv3
2	Resnet.layer2.3.conv3	Resnet.layer2.3.conv3
3	Resnet.layer3.22.conv3	Resnet.layer3.5.conv3
4	Resnet.layer4.2.conv3	Resnet.layer4.2.conv3

The criteria of the selected layer is the last convolution layer of the last block in the 4 layers of ResNet model.

5. Experimental Results

5.1. Performance Comparison

To validate the performance of the proposed method, two experiments are carried out and the results are presented with the mean and standard deviation after 5 identical experiments. Two experiments are performed using the five types of dataset described in section 4.1.

The first experiment compares the proposed method with five existing methods: L^2, $L^2 - FE$ [29], $L^2 - SP$ [21], DELTA [29], and AFA [32]. For all datasets, the proposed regularization method shows an overall improved performance compared to previous transfer learning regularization methods. The classification accuracy for the CUB-200-2011 dataset [37] was boosted by approximately 0.32% in comparison to the AFA method. For the MIT Indoors 67 [38], the improvement is 0.48% is, which is quite good for a small amount of training data. However, the improvement is relatively small for the Stanford Dogs 120 [35] and Caltech 256-60 [36]. The first experimental results can be confirmed

 doi:10.20944/preprints202404.1423.v1

by Table 4. In the second experiment, we compared the performance between SPA and GCA, which are submodules of the proposed method, and AST and ACT, which are submodules of the AFA method. After applying each submodule to the target model one by one, SPA and GCA showed similar or improved classification accuracy results across the board. For most datasets, there was a slight improvement of the SPA over the AST. Regarding the Caltech 256-60, the GCA showed about 0.85% improvement over the ACT. However, for the MIT Indoors 67 dataset, the SPA resulted in 0.18% less accuracy, whereas the GCA yielded an increase by 0.69% over the ACT. The results can be checked through Table 5. In addition, both proposed submodules converged faster than the submodules of the previous AFA method. The SPA module converged quickly but had similar performance, while the GCA module converged faster and had improved performance. The comparison of the convergence speed can be confirmed by Figure 2.

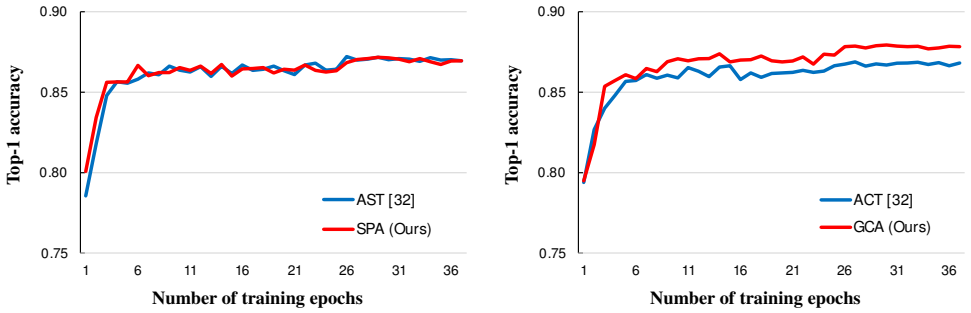
Table 4. Comparison of top-1 accuracy (%) results with different methods on five datasets

Model	Dataset	Methods					
		L^2	$L^2 - FE$ [29]	$L^2 - SP$ [21]	DELTA [29]	AFA [32]	Proposed
ⓐ	①	82.37% ± 0.25	87.66% ± 0.15	87.28% ± 0.17	87.48% ± 0.11	88.22% ± 0.07	88.44% ± 0.04
	②	78.40% ± 0.10	62.09% ± 0.07	79.89% ± 0.09	80.04% ± 0.13	80.30% ± 0.03	80.62% ± 0.08
	③	84.39% ± 0.41	83.18% ± 0.13	85.72% ± 0.10	85.49% ± 0.16	86.15% ± 0.06	86.42% ± 0.01
	④	86.69% ± 0.23	83.64% ± 0.10	87.72% ± 0.15	86.94% ± 0.07	87.83% ± 0.03	87.94% ± 0.06
ⓑ	⑤	83.06% ± 0.19	82.00% ± 0.10	83.70% ± 0.17	84.06% ± 0.08	84.34% ± 0.10	84.82% ± 0.13

Model ⓐ : ResNet-101, ⓑ : ResNet-50
Dataset ① : Stanford Dogs 120, ② : CUB-200-2011, ③ : Caltech 256-30, ④ : Caltech 256-60, ⑤ : MIT Indoor 67

Table 5. Comparison of top-1 accuracy (%) results with different submodules on five datasets

Model	Dataset	Modules			
		AST [32]	ACT [32]	SPA	GCA
ResNet-101	Stanford Dogs 120	88.12% ± 0.08	88.06% ± 0.04	88.29% ± 0.07	88.18% ± 0.04
	CUB-200-2011	80.37% ± 0.16	80.29% ± 0.12	80.47% ± 0.14	80.48% ± 0.15
	Caltech 256-30	85.89% ± 0.06	85.69% ± 0.07	85.98% ± 0.09	86.14% ± 0.09
	Caltech 256-60	87.20% ± 0.12	87.02% ± 0.19	87.30% ± 0.11	87.87% ± 0.08
ResNet-50	MIT Indoor 67	84.36% ± 0.09	84.03% ± 0.14	84.18% ± 0.13	84.72% ± 0.06



(a) Comparison between AST and SPA modules (b) Comparison between ACT and GCA modules

Figure 2. Comparison of top-1 accuracy results with different submodules on Caltech 256-60

5.2. *Ablation Study*

We conducted an experiment to assess how the reduction rate of GCA submodules, expressed as r, impacts the regularisation model. The reduction rate is defined in section 4.2. Using the ResNet-101

 doi:10.20944/preprints202404.1423.v1

model and the Caltech 256-30 dataset, we measured the object classification accuracy while varying the r value of GCA, and the experimental results are shown in Table 6. Experiments have shown that classification accuracy tends to decrease as r increases. This is because more information contained in the filter values in the source layers is lost at the bottleneck of the fully connected layer. In this paper, we set the reduction rate to 4, which is the lowest level.

Another experiment was conducted to determine the difference in transfer learning performance based on the training order of the SPA and GCA submodules. The comparison of classification accuracy using the Caltech 256-60 dataset is shown in Table 7. Better result is obtained when the SPA submodule is trained after the GCA module. To reflect this result, we trained SPA for the first half of the total training session and then GCA for the remainder of the session.

Table 6. Effects of reduction rate r on global channel attention module

Model	Dataset	Reduction rate r			
		4	8	16	32
ResNet-101	Caltech 256-30	$86.14\% \pm 0.09$	$86.06\% \pm 0.04$	$86.06\% \pm 0.21$	$86.05\% \pm 0.16$

Table 7. Effects of training order of proposed submodules

Model	Dataset	Methods	
		SPA $\rightarrow$ GCA	GCA $\rightarrow$ SPA
ResNet-101	Stanford Dogs 120	$88.44\% \pm 0.04$	$88.23\% \pm 0.13$
	CUB-200-2011	$80.62\% \pm 0.08$	$79.93\% \pm 0.05$
	Caltech 256-30	$86.42\% \pm 0.01$	$86.31\% \pm 0.08$
	Caltech 256-60	$87.94\% \pm 0.06$	$88.05\% \pm 0.08$
ResNet-50	MIT Indoor 67	$84.82\% \pm 0.13$	$84.30\% \pm 0.07$

6. Conclusions

In this paper, we propose an improved deep transfer learning regularization method. The proposed method uses a global channel attention submodule that determines the channel importance of all layers, unlike existing methods that use channel importance only within a single layer. Furthermore, the proposed self-pixel attention submodule uses both the feature maps of the source and target models, unlike existing methods that only utilize the feature map of the source model, so that target feature information can also be considered. The performance of the novel attention submodules has improved both in terms of classification accuracy and training convergence speed. In the future, the proposed method can be extended to an improved transfer learning regularization method based on knowledge distillation through a method of local selection of feature maps of the spatial attention module.

Author Contributions: Conceptualization, Changhee and Sang-ug; methodology, Chanhee; software, Changhee; validation, Sang-ug; formal analysis, Changhee; investigation, Sang-ug; writing—original draft preparation,Changhee; writing—review and editing, Sang-ug; visualization, Changhee; supervision, Sang-ug; project administration, Sang-ug.; funding acquisition, Sang-ug. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Research Foundation of Korea grant funded by Korea government(NRF-2022R1A2C1004674).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The pre-trained models using ImageNet and Places 365 source data for transfer learning are available at pytorch pre-trained models and https://github.com/CSAILVision/places365, respectively.

 doi:10.20944/preprints202404.1423.v1

The dataset on Stanford Dogs 120 is available at http://vision.stanford.edu/aditya86/ImageNetDogs/. The Caltech-UCSD Birds-200-2011 (CUB-200-2011) and Caltech 256 dataset are available at https://www.vision.caltech.edu/datasets/. The MIT Indoor 67 dataset is available at https://web.mit.edu/torralba/www/indoor.html.

Acknowledgments: This work is supported by the National Research Foundation of Korea grant funded by Korea government (NRF-2022R1A2C1004674).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Hussein, S.; Kandel P.; Bolan, C.W.; Wallace, M.B.; Bagci, U. Lung and pancreatic tumor characterization in the deep learning era: novel supervised and unsupervised learning approaches. IEEE Transactions on Medical Imaging 2019, 38, 1777-1787.
2. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-shot text-to-image generation. In Proceedings of the International Conference on Machine Learning 2021, 139, 8821-8831.
3. Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Gläser, C.; Timm, F.; Wiesbeck, W.; Dietmayer, K. Deep multi-modal object detection and semantic segmentation for autonomous driving: datasets, methods, and challenges. IEEE Transactions on Intelligent Transportation Systems 2020, 22, 1341-1360.
4. Kang, C.; Kang, S.-u. Self-supervised denoising image filter based on recursive deep neural network structure. Sensors 2021, 21, 7827.
5. Orabona, F.; Jie, L.; Caputo, B. Multi Kernel Learning with online-batch optimization. Journal of Machine Learning Research 2012, 13, 227-253.
6. Nilsback, M.-E.; Zisserman, A. Automated flower classification over a large number of classes. In Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing 2008, 722-729.
7. Cui, Y.; Song, Y.; Sun, C.; Howard, A.; Belongie, S. Large scale fine-grained categorization and domain-specific transfer learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, 4109-4118.
8. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2009, 248–255.
9. 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 2016, 770–778.
10. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 2014.
11. Wang, Y.-X.; Ramanan, D.; Hebert, M. Growing a brain: Fine-tuning by increasing model capacity. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, 2471-2480.
12. Guo, Y.; Shi, H.; Kumar, A.; Grauman, K.; Rosing, T.; Feris, R. Spottune: transfer learning through adaptive fine-tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, 4805-4814.
13. Tan, T.; Li, Z.; Liu, H.; Zanjani, F.G.; Ouyang, Q.; Tang, Y.; Hu, Z.; Li, Q. Optimize transfer learning for lung diseases in bronchoscopy using a new concept: Sequential fine-tuning. IEEE Journal of Translational Engineering in Health and Medicine 2018, 6, 1-8.
14. Ge, W.; Yu, Y. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, 1086-1095.
15. Ng, H.-W.; Nguyen, V.D.; Vonikakis, V.; Winkler, S. Deep learning for emotion recognition on small datasets using transfer learning. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction 2015, 443-449.
16. Zhao, W. Research on the deep learning of the small sample data based on transfer learning. In Proceedings of the AIP Conference 2017, 1864, 020018.
17. Mohammadian S.; Karsaz, A.; Roshan, Y.M. Comparative study of fine-tuning of pre-trained convolutional neural networks for diabetic retinopathy screening. In Proceedings of the 2017 24th National and 2nd International Iranian Conference on Biomedical Engineering 2017, 1-6.

18. Pratt, H.; Coenen, F.; Broadbent, D.M.; Harding, S.P.; Zheng, Y. Convolutional neural networks for diabetic retinopathy. Procedia Computer Science 2016, 90, 200-205.
19. Wang, Z.; Dai, Z.; Poczos, B.; Carbonell, J. Characterizing and avoiding negative transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, 11293-11302.
20. Zhao, Z.; Zhang, B.; Jiang, Y.; Xu, L.; Li, L.; Ma, W.-Y. Effective domain knowledge transfer with soft fine-tuning. arXiv preprint arXiv:1909.02236 2019.
21. Li, X.; Grandvalet, Y.; Davoine, F. Explicit inductive bias for transfer learning with convolutional networks. In Proceedings of the 35th International Conference on Machine Learning 2018, 80, 2825-2834.
22. Li, X.; Grandvalet, Y.; Davoine, F. A baseline regularization scheme for transfer learning with convolutional neural networks. Pattern Recognition 2020, 98, 107049.
23. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 2015.
24. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. FitNets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 2014.
25. Zhang, Y.; Xiang, T.; Hospedales, T.M.; Lu, H. Deep mutual learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, 4320-4328.
26. Mirzadeh, S.I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In Proceedings of the AAAI Conference on Artificial Intelligence 2020, 34, 5191-5198.
27. Li, T.; Li, J.; Liu, Z.; Zhang, C. Few sample knowledge distillation for efficient network compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, 14639-14647.
28. Li, Z.; Hoiem, D. Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017, 40, 2935-2947.
29. Li, X.; Xiong, H.; Wang, H.; Rao, Y.; Liu, L.; Chen, Z.; Huan, J. Delta: Deep learning transfer using feature map with attention for convolutional networks. arXiv preprint arXiv:1901.09229 2019.
30. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
31. Mnih, V.; Heess, N.; Graves, A.; kavukcuoglu, K. Recurrent models of visual attention. In Proceedings of the Neural Information Processing Systems 2014, 2204-2212.
32. Xie, Z.; Wen, Z.; Wang, Y.; Wu, Q.; Tan, M. Towards effective deep transfer via attentive feature alignment. Neural Networks 2021, 138, 98-109.
33. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, 7132-7141.
34. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017, 40, 1452–1464.
35. Khosla, A.; Jayadevaprakash, N.; Yao, B.; Li, F.-F. Novel dataset for fine-grained image categorization: Stanford dogs. In Proceedings of the First Workshop on Fine-Grained Visual Categorization in IEEE Conference on Computer Vision and Pattern Recognition 2011, 1-2.
36. Griffin, G.; Holub, A.; Perona, P. Caltech-256 object category dataset. California Institute of Technology 2007.
37. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. The caltech-ucsd birds-200-2011 dataset. California Institute of Technology 2011.
38. Quattoni, A.; Torralba, A. Recognizing indoor scenes. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition 2009, 413–420.