AFINETS: ATTENTIVE FEATURE INTEGRATION NETWORKS FOR IMAGE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Convolutional Neural Networks (CNNs) have achieved a tremendous success in a number of learning tasks, e.g., image classification. Recent advances in CNNs, such as ResNets and DenseNets, mainly focus on the skip and concatenation operators to avoid gradient vanishing. However, such operators largely neglect information across layers (as in ResNets) or involve tremendous redundancy of features repeatedly copied from previous layers (as in DenseNets). Furthermore, due to the quadratic complexity of memory usage, many deep residual-like networks are obliged to abandon concatenation connections. In this paper, we design Attentive Feature Integration (AFI) modules, which can be applicable to most recent network architectures , leading to new architectures named as AFINets. AFINets can by adaptively integrates distinct information through explicitly modeling the subordinate relationship between different levels of features. Experimental results on benchmark datasets have demonstrated the effectiveness of the proposed AFI modules.

1 INTRODUCTION

Convolutional Neural Networks(CNNs) have achieved the state-of-the-art results in a variety of computer vision tasks (Wang et al., 2017; Jiang et al., 2013; Gatys et al., 2016). Recently, feature reuse (Huang et al., 2017; Lee et al., 2019), addition and concatenation mechanisms, fuels the further progress in the performance of CNNs due to the improvement of utilization of features and the representation ability of models. The key insight of feature reuse is that the features from different layers in CNN contain distinct information. More specifically, the features of shallow layers contain more space information but relatively less advanced semantic information while the features of deep layers contain more advanced semantic information but relatively less space information. For instance, bar detectors in early layers might localize bars precisely, but cannot discriminate whether the bars are desk legs or crutches. Additionally, Morcos et al. (2018) shows that the class selectivity, which is proved to be related with the generalization gap to some degree, is prone to increase with depth. This indicates that reusing low-level features is in favor of closing generalization gap. These observations suggest that aggregation of multiple levels of features is vital to reason.

Addition and concatenation are two fundamental aggregation operations of feature reuse for designing network architecture. From the perspective of addition, the shortcut connection can alleviate the problem of gradient vanishing and smooth loss landscapes without extra memory usage (Li et al., 2018a). Compared with addition, aggregating the features from layers by concatenation maintains the most merits of addition, assists to diversify depth and leads to implicit deep supervision (Huang et al., 2017). However, concatenation is abandoned by many deep residual-like networks for the quadratic complexity of memory usage. Another shortcoming of concatenation is that, for the reason that features are entangled with each other, model training suffers from concatenated features which are indiscriminately selected.

To address these two problems, we explicitly model a lightweight and selective feature integration scheme for better feature reuse, leading to the Attentive Feature Integration (AFI) module, as illustrated in Figure 1. Firstly, for several successive feature maps (i.e., raw features) with C channels, each feature is extracted to a vector by a squeeze operation that captures information from a large spatial extent. Thus, the global context is embedded in the vectors. Secondly, the vectors are scored and normalized sequentially by the attention mechanism (which consists of the shared scoring

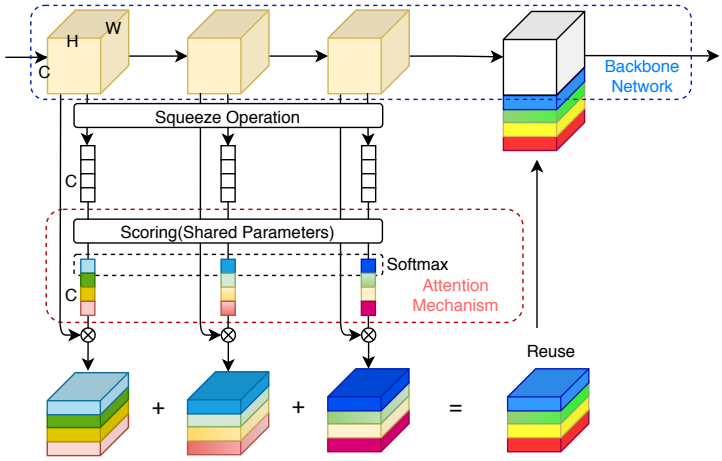


Figure 1: The AFI block can automatically extract important features for later feature reusing based on the learned features.

function and the channel-by-channel softmax function) in order to re-calibrate features. At last, we obtain the resulting feature via a summation of re-calibrated features, each channel of which can be viewed as a convex combination over the raw features.

Different from concatenation that exploits whole features information with heavy computational cost (as used in DenseNets), the memory usage incurred by the AFI module has linear rather than quadratic complexity. Hence, the AFI module is much more efficient to be plugged into residual-like networks. Notably, the plugged networks are also beneficial from deep supervision and diversified depth, leading to better accuracy with less parameters and FLOPs. Furthermore, the experimental results show that our feature-wise integration module is compatible with the self-attention mechanism (e.g., SE module (Hu et al., 2018b)), which models inter-dependencies of feature between channels. Moreover, through the ablation studies, we discuss the interpretation of the AFI module. Equipped with the AFI module, the corresponding AFI-ResNet-152 increases the Top-1 accuracy rate of the vanilla ResNet-152 on ImageNet by 1.24%, with about 10% fewer FLOPs and 9.2% reduction of parameters.

Our main contributions can be summarized as follows:

- **Lightweight and Plug-and-Play:** The AFI module is lightweight, which avoids the quadratic complexity of memory usage. Consequently, it is easy to apply the AFI module into CNNs. For example, AFI-ResNet can be constructed by shrinking convolutional operations and plugging the AFI module into ResNet (He et al., 2016). AFI-MobileNetV2, AFI-ShuffleNetV2, and AFI-ResNeXt can be composed similarly.
- **Interpretability:** Through ablation study, Figure 2 shows that the borderline of the target is sharper by applying the AFI module. Moreover, the features applied by the AFI module are less class-conditional than the vanilla, which implicates to less generalization gap.
- **Higher Accuracy and Lower FLOPs:** Experimental results show that our AFI module significantly improves the representational power of the network. For example, AFI-ResNet-152 increases the Top-1 accuracy rate in ImageNet by 1.24% compared to ResNet-152 while decreases the FLOPs by about 10% and the number of parameters by about 9.2%.

2 Related works.

Modern CNN Architectures. Deep convolutional neural networks(CNNs) have dominated image classification since the AlexNet (Krizhevsky et al., 2012) and VGG-Net (Simonyan & Zisserman, 2014) are proposed. After that, substantial efforts have been made to improve the efficiency of CNNs. The modular design strategy in GoogleNet (Szegedy et al., 2015) simplifies the network

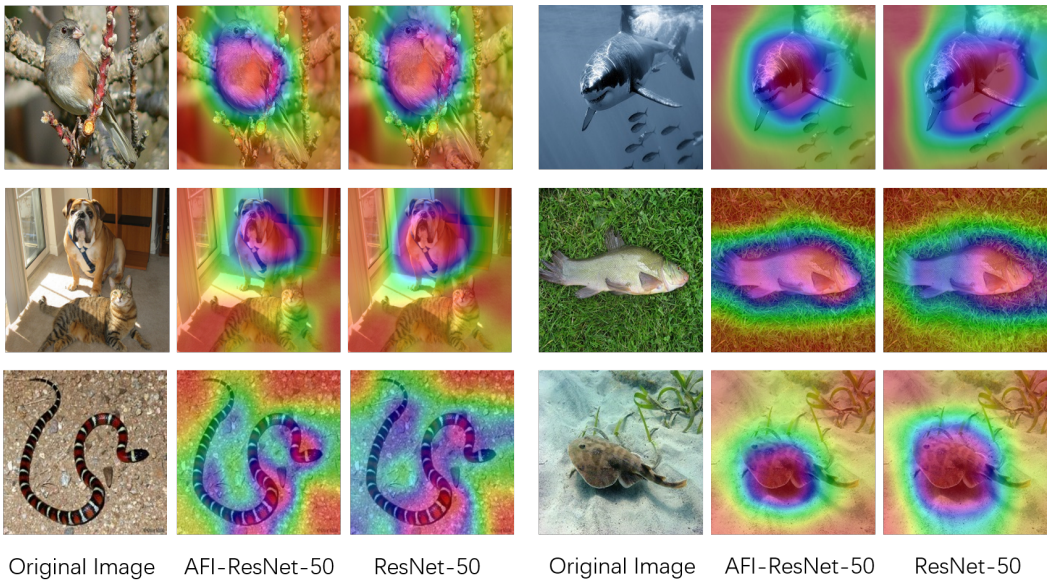


Figure 2: Illustration of the impact of the AFI module. All original images come from the ImageNet dataset. Heatmaps generated by Grad-CAM (Selvaraju et al., 2017) illustrates which areas the network pays more attention to. Compared to the vanilla ResNet-50, the area that AFI-ResNet-50 paid most attention to is much smaller.

architecture and the multi-path structure in each module shows a great success. ResNet introduces the shortcut connection alleviating the difficulty in deep network training. DenseNet (Huang et al., 2017) densely connects all preceding layers to take full advantage of preceding feature maps. Based on those fundamental architectures, some advanced variants(e.g., ResNeSt (Zhang et al., 2020)) have been proposed and have achieved impressive performance in many computer vision tasks.

Feature Reuse. Maximizing the feature utilization is one way to make the CNNs more efficient. The feature reuse mechanism in ResNet summarizes the whole output features of the frontier blocks and also replaces the identity mapping into residual mapping by short connections. Due to its simple structure and outstanding performance, a series of variants have been proposed, including ShuffleNet (Ma et al., 2018), ResNeXt (Xie et al., 2017) and Inception-ResNet (Szegedy et al., 2017). At the same time, DenseNet (Huang et al., 2017) demonstrates that reusing feature by densely concatenating all the features in frontier layers can effectively alleviate the difficulty of training and improve the network performance with an increased risk of over-fitting. VoVNet (Lee et al., 2019), VoVNetV2 (Lee & Park, 2019) overcomes the inefficiency of dense connection by concatenating all features only once in the last feature map and achieve the state-of-the-art performance in instance segmentation. As the trade-offs between ResNets (He et al., 2016) and DenseNets (Huang et al., 2017), Dual Path Network (Chen et al., 2017) and Mixed link Network (Wang et al., 2018a) combine the two forms of connection, addition and concatenation.

Attention Mechanisms. The benefits of attention mechanism have been demonstrated across a range of tasks. Squeeze-and-Excitation block (Hu et al., 2018b) highly appreciates attention mechanism and thus well improves the accuracy of varied CNNs. They use global average-pooled features to exploit the inter-channel relationship and to compute the channel-wise attention. Besides, there are several other researches to utilize the attention mechanism and improve the results of CNNs in various vision tasks. CBAM (Woo et al., 2018) further add the spatial attention to the SE modeule and results in better plug-and-play modules. Wang et al. (2017) propose soft mask branches to refine the feature maps by adding attention knowledge. Non-local Neural Networks (Wang et al., 2018b) proposes non-local module to integrate the global attention information. Libra R-CNN (Pang et al., 2019) designs the balanced feature pyramid which refine the semantic feature from multi-level features. BASNet (Qin et al., 2019) pays more attention to the boundary of the mask by the boundary-aware loss function. However, few works focus on the mix of attention mechanism and feature reuse. The proposed AFI module thus aims to improve feature reuse by the attention mechanism.

3 OUR MODEL

In this section, firstly, we introduce our Attentive Feature Integration (AFI) module. Secondly, we compare our module with previous works such as DenseNet combined with the self-attention module and discuss the way to implement the AFI module for reusing features. At last, for validating the capacity of the proposed model, we integrate AFI blocks with ResNet (He et al., 2016) and MobileNetV2 (Sandler et al., 2018) to form AFI-ResNet and AFI-MobileNetV2, whose details of the architectures will be presented.

3.1 ATTENTIVE FEATURE INTEGRATION MODULES

The squeeze operations and the attention mechanism compose our AFI modules. The attention mechanism is a selective aggregation of information so that the resulting features can be easily exploited by subsequent transformations.

Formally, for N different level features $\mathbf{X}_i \in \mathbb{R}^{H\times W\times C}, i \in \{1, 2, ..., N\}$, the result feature $\mathbf{R} \in \mathbb{R}^{H\times W\times C}$ is gotten by:

$$\begin{aligned}
&\mathbf{z}_i = F_{sq}(\mathbf{X}_i), \quad \mathbf{s}_i = F_{sc}(\mathbf{z}_i, \mathbf{W}), \\
&\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N], \quad \widetilde{\mathbf{S}} = [Softmax(\mathbf{s}^T_{*,1}), \cdots, Softmax(\mathbf{s}^T_{*,C})]^T, \\
&\mathbf{R} = \widetilde{\mathbf{s}}_1 \otimes \mathbf{X}_1 + \widetilde{\mathbf{s}}_2 \otimes \mathbf{X}_2 + \cdots + \widetilde{\mathbf{s}}_N \otimes \mathbf{X}_N.
\end{aligned} \tag{1}$$

Here $F_{sq}(\cdot)$ denotes a squeeze function(e.g., global average pooling), which gathers contextual long-range feature interactions, embedding global context into a vector descriptor. By shrinking $\mathbf{X}_i$ on its spatial dimensions $H \times W$, the channel-wise statistic $\mathbf{z}_i \in \mathbb{R}^C$ is generated, where C is the number of channels. On the second stage, the shared scoring function $F_{sc}(\cdot)$ is applied into vector descriptors to produce an embedding of importance $\mathbf{s} \in \mathbb{R}^C$. $F_{sc}(\cdot)$ is set as two transformation matrices around the activation function:

$$\mathbf{s}_i = F_{sc}(\mathbf{z}_i, \mathbf{W}) = \mathbf{W}_2 ReLU(\mathbf{W}_1 \mathbf{z}_i), \tag{2}$$

where the vector $\mathbf{s}_i$ is parameterized by forming a bottleneck layer with weights $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r}\times C}$, a dimensionality-increasing layer with weights $\mathbf{W}_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ and the reduction ratio r. Next, a matrix $\mathbf{S} \in \mathbb{R}^{N\times C}$, $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N]$ is obtained by concatenating all the N vectors. Then, the matrix $\widetilde{\mathbf{S}} \in \mathbb{R}^{N\times C}$ are obtained by normalizing $\mathbf{S}$ along the N dimension using a Softmax function. At last, the $\otimes$ notation denotes the channel-wise multiplication. The result feature $\mathbf{R}$ is the combination of all the re-weighted input features, $\widetilde{\mathbf{s}}_i \otimes \mathbf{X}_i$.

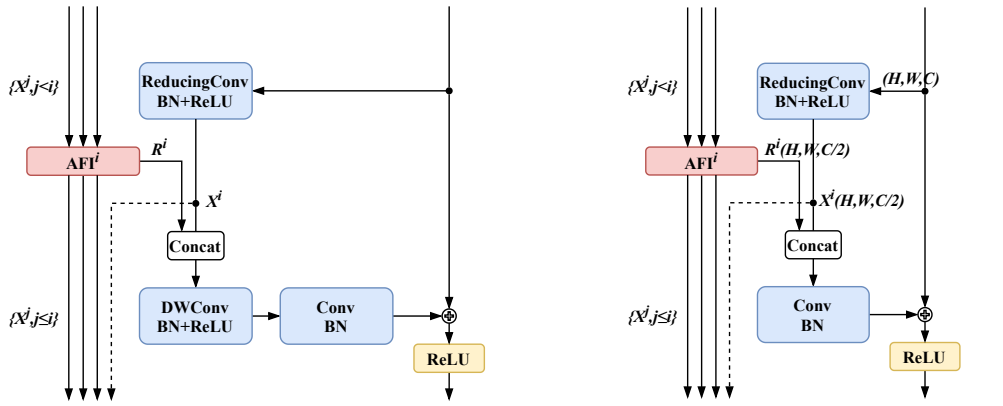
3.2 COMPARISON WITH PREVIOUS WORKS

Recently, a series of literature attempts to incorporate the attention mechanism to improve the performance of CNNs. One of the most popular computational units is the Squeeze-and-Excitation(SE) module introduced in Hu et al. (2018b). Compared to the SE module, our AFI module focus on explicitly modeling the feature integration instead of channel-wise selection. Moreover, compared to DenseNet with the self-attention module like CAPR-DenseNet (Zhang et al., 2019), our module can be applied to deeper and larger network, while avoiding quadratic complexity memory usage and running time by substituting the independent exciter for the shared attention mechanism. Compared to the SKNet (Li et al., 2019) that selects the efficient kernel size, our AFI module can utilize different level(e.g., positional and semantic) information by reusing features.

3.3 AFI MODULES FOR FEATURE REUSE

In order to better describe our method, we have followed the definition of *stage* and *building blocks* in the previous work (Hu et al., 2018a). More specifically, a *stage* consists of several *building blocks* with the same shape of feature stacking repeatedly.

The AFI module is a supplement to backbone networks and is simple to be applied to the residual-like networks. For building block i of each stage, the features, $X_1, X_2, \cdots, X_{i-1}$, in previous building blocks are the inputs of the AFI module. The output of the AFI module, the fused feature



(a) The AFI-MobileNet building block (b) The AFI-ResNet building block

Figure 3: The architecture of the AFI-MobileNet building block is shown in Figure 3(a). The architecture of the AFI-ResNet building block is shown in Figure 3(b).

Table 1: The accuracy rate (%) comparison of applying our AFI module to different residual stages of ResNet-32 and ResNet-110. Best results are marked in bold.

AFI Stage	ResNet-32	Stage 1	Stage 2	Stage 3
C100	71.16	**71.38**$_{(+0.22)}$	70.63$_{(-0.53)}$	70.75$_{(-0.41)}$
AFI Stage	ResNet-110	Stage 1	Stage 2	Stage 3
C100	73.73	**75.03**$_{(+1.30)}$	74.13$_{(+0.40)}$	74.02$_{(+0.29)}$

R^i, will be concatenated with feature X_i and fed into the subsequent convolution layer. More details of AFI-Networks are shown in Figure 3.

Two different settings of the AFI-Networks are used in this paper. The first setting is that replacing half of the output from first convolution operations in residual building blocks with the output of the AFI module as shown in Figure 3(b) and the reduction ratio r is 4. We refer a network with the first setting to an AFI-Network. Another setting remains the original network architecture and expands the second convolution. In this setting, we set the reduction ratio r to 16. For example, the second convolution kernel of the basic building block in ResNet is changed into $\mathbb{R}^{k\times k\times 2C\times C}$. We name it AFI-Network-B. It is worth mentioning that the only difference between the two settings is the width of the network.

To further illustrate the implementation of our module, we take the AFI-ResNet and the AFI-MobileNetV2 building blocks as examples. The diagram in Figure 3 shows details of the proposed AFI-ResNet and AFI-MobileNetV2. We concatenate the result feature of the AFI module with the output feature of the first convolutional layer. As many building blocks (e.g., the bottleneck building block, the MobileNetV2 (Sandler et al., 2018) block) use first 1×1 convolution to reduce parameters, our AFI modules can neutralize the impairment of bottleneck convolution coming from the reduction of channels.

4 Experiment

Before presenting results on real-world datasets, we first show the ablation study on the settings of the AFI modules.

The Role of AFI Module To illustrate whether the low-level features are collected by our AFI module, we generate the heatmap by Grad-CAM (Selvaraju et al., 2017). As we can see in the Figure 2, our network has more details of detected things, and meanwhile, the heatmap generated by ResNet-50 is too smooth to draw the specific borderline of detected things. With the help of

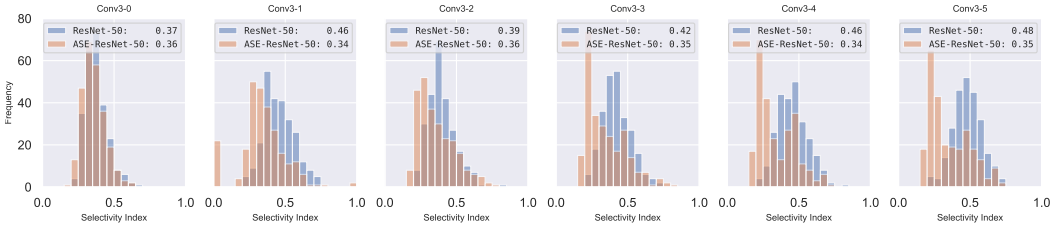


Figure 4: Each figure depicts the class selectivity index distribution for features in both the baseline ResNet-50 and AFI-ResNet-50 various building blocks in the $Conv3$ stage of their architectures. The distributions come from the output of the AFI module and corresponding position of the vanilla in each building block. The mean value is reported after the label.

Table 2: The accuracy rate(%) of AFI-MobileNetV2 with various of the self-attention model in CIFAR-100.

MobileNetV2	Baseline	AFI	AFI-SE	AFI-GE	AFI-CBAM	AFI-ECA
C100	75.20	75.94	$76.15_{(+0.21\%)}$	$76.22_{(+0.28\%)}$	$76.51_{(+0.57\%)}$	$76.76_{(+0.82\%)}$

low-level feature reuse, the tiny borderlines are more shown in the figure. Besides, compared to ResNet-50, the area that AFI-ResNet-50 paid most attention to decreases obviously.

Besides, the class selectivity index metric introduced by Morcos et al. (2018) to analyze the semantic meaning of features. This metric computes, for each feature map, the difference between the highest class-conditional mean activity and the mean of all remaining class-conditional activities over the testing dataset. The measurement is normalized between zero and one where one indicates that a filter only fires for a single class and zero indicates that the filter produces the class-agnostic value. The less class selectivity index contend to the more generalization of channels in a degree. As shown in figure 4, the AFI-ResNet-50 is able to learn tinier and more generalized features instead of features of a specific class than the vanilla in $Conv3$ stage. With the help of generalized features, the generalization gap between the training and the testing dataset will be closed.

The Position of AFI Module In this ablation study, we study whether low-level feature maps or high-level feature maps are more hospitable for feature reuse by the AFI module. Previous study Yosinski et al. (2014) shows that the feature maps learned by the earlier convolutional layers are more general. According to Table 1, by only applying our AFI module to the stage 1 of ResNet-32 or ResNet-110, the accuracy rate increases obviously, which means low-level features have more practical impacts.

Potential in Width Networks Tables 4, 5 show our AFI module is adapted for increasing the width of networks. The results of AFI-Network-B show that our AFI module offers a trade-off between improved accuracy and increased model complexity for the real situations. For instance, for a mobile user, the company can adopt AFI-Networks with lower computational complexity while a computer user can adopt AFI-Network-Bs with higher accuracy.

Compatibility with Other Self-attention Models Intuitively, the AFI-module is a feature-wise attention mechanism; alternatively, the self-attention module aims to model interdependence between channels explicitly. We further conducted experiments on CIFAR-100 to demonstrate compatibility. In the experiments, we utilize the self-attention modules(SE(Hu et al., 2018b), GE(Hu et al., 2018a), CBAM(Woo et al., 2018), ECA(Wang et al., 2020)) to the resulting features. As shown in Table 2, all the AFI-SE, AFI-GE, AFI-CBAM, and AFI-ECA$_{(+0.82\%)}$ models get better results, which indicates that our model is compatible with other self-attention modules.

4.1 Experiments on CIFAR

The CIFAR (Krizhevsky et al., 2009) dataset consists of 60,000 RGB pictures, each with a size of 32×32. 50,000 of them are used as the training set and 10,000 are used for testing. The CIFAR-10 mission requires the network to correctly classify the pictures into 10 categories, such as airplanes

Table 3: The architecture details of AFI-ResNet-6N+2 for CIFAR dataset. The operations and feature shapes are listed inside the brackets and the number of stacked blocks is shown outside.

Name	Output Size	AFI-ResNet-(6N+2)
Conv_0	32×32	3×3, 16
Conv_1	32×32	$\begin{bmatrix} \text{AFI +} & 3\times3, 16 \\ & 3\times3, 16 \end{bmatrix} \times N$
Conv_2	16×16	$\begin{bmatrix} \text{AFI +} & 3\times3, 32 \\ & 3\times3, 32 \end{bmatrix} \times N$
Conv_3	8×8	$\begin{bmatrix} \text{AFI +} & 3\times3, 64 \\ & 3\times3, 64 \end{bmatrix} \times N$
	num_class	AP,FC,Softmax

Table 4: Accuracy rates (%) on CIFAR-100 dataset. All results are reproduced by ourselves for a fair comparison. Our network results are bold in the table. The FLOPs are calculated by assuming the batch size of 32.

Model	C100	Params	FLOPs
ResNet-32 (He et al., 2016)	71.16	472.76K	2.23G
AFI-ResNet-32	**71.09**	378.23K	1.78G
AFI-ResNet-32-B	**72.57**	668.98K	3.15G
ResNet-110 (He et al., 2016)	73.73	1.74M	8.17G
AFI-ResNet-110	**74.03**	1.35M	6.23G
AFI-ResNet-110-B	**75.69**	2.57M	12.06G
ShuffleNetV2 (Ma et al., 2018)	70.71	0.94M	1.32G
AFI-ShuffleNetV2	**72.06**	0.95M	1.32G
AFI-ShuffleNetV2-B	**72.24**	1.29M	1.76G
MobileNetV2 (Sandler et al., 2018)	75.20	2.41M	3.03G
AFI-MobileNetV2	**75.94**	2.25M	2.67G
AFI-MobileNetV2-B	**77.08**	2.84M	3.16G
ResNext-29(32×4d) (Xie et al., 2017)	78.44	4.87M	24.95G
AFI-ResNext-29(32×4d)	**79.37**	4.26M	21.74G
AFI-ResNext-29(32×4d)-B	**79.85**	5.23M	25.90G

and automobiles. CIFAR-100 requires the network to classify pictures into 100 categories. We train our network on the training dataset and evaluate it on the test dataset.

By integrating the AFI module with ResNet, ShuffleNetV2, MobileNetV2 and ResNext, we get their AFI counterparts. All the backbone networks have a residual mapping. So we apply the AFI module to the first convolution layer in building blocks to avoid affecting residual mapping and meanwhile neutralize the wastage of the bottleneck convolution. Table 3 shows AFI-ResNet architecture. By setting $N = 5$ and $N = 18$ separately, AFI-ResNet-32 and AFI-ResNet-110 are acquired.

In this experiment, we use SGD with a momentum of 0.9 and a weight decay of 1e-4. We train the networks with the batch size to 64 for 300 epochs. The learning rate is initial to 0.1 and divided by 10 at 50%, 75% of training process, respectively. Proportion is adopted in Huang et al. (2017). Data augmentation(mirroring/shifting) is used in training. Because MobileNetV2 (Sandler et al., 2018), ShuffleNetV2 (Ma et al., 2018), and other networks are not designed for the CIFAR dataset, we adopt their variants from Github[1]. All results are reproduced by ourselves and use the same experiment settings are adopted for a fair comparison.

Table 4 shows the comparison of classification error between the original networks and their corresponding AFI counterparts. As we can see, most of networks work better in classification with assistance of our AFI module. For example, AFI-ResNext29(32×4d) increases the accuracy rate by 0.93% compared with ResNext29(32×4d) (Xie et al., 2017) on the CIFAR-100 dataset. Besides, as

[1]https://github.com/kuangliu/pytorch-cifar

Table 5: The table shows the accuracy rates (%) of networks on the ImageNet validation set. Our results are marked in bold. All results are reproduced for a fair comparison. The FLOPs are calculated by assuming the batch size of 32.

Model	Params(M)	FLOPs(G)	Top-1 Prec.	Top-5 Prec.
ResNet-50 (He et al., 2016)	25.56	131.57	75.3	92.2
AFI-ResNet-50	**23.85**$_{(-1.71)}$	**121.72**$_{(-9.85)}$	**76.19**$_{(+0.89)}$	**92.88**$_{(+0.68)}$
AFI-ResNet-50-B	**33.86**	**176.05**	**77.02**	**93.55**
ResNet-152 (He et al., 2016)	60.19	369.88	77.0	93.3
AFI-ResNet-152	**54.67**$_{(-5.52)}$	**332.24**$_{(-37.64)}$	**78.24**$_{(+1.24)}$	**93.98**$_{(+0.68)}$
SE-ResNet-50	28.07	131.83	76.71	93.38
SE-ResNet-152	66.77	370.52	78.43	94.27

Table 6: The comparison of the memory usage of AFI-ResNet-50 and ResNet-50 in the training and testing process. The batch size is 64.

Model	ResNet-50	AFI-ResNet-50
Training/Testing Memory (MiB)	7447/3827	**7001/2671**

shown in the results, other AFI-Networks also increase the accuracy rates while decrease or remain at least the number of parameters and FLOPs.

4.2 Experiments on ImageNet

The effect of the AFI module is also evaluated on the ImageNet 2012 dataset (Simonyan & Zisserman, 2014) which composes about 1.3 million training images and 50k validation images. Both top-1 and top-5 classification accuracy rates are reported on the validation dataset.

In this experiment, we use SGD with a momentum of 0.9 and a weight decay of 1e-4. We train the networks with batch size 64 for 90 epoch. The initial learning rate is 0.1 * batch_size / 256 and divided by 10 at 30, 60, and 80 epochs, respectively. 224×224 images serving as the inputs of the network are cropped from the resized raw images or their horizontal flips. Data augmentation in Li et al. (2018b) is used in training. We evaluate our model by applying a center-crop with 224×224.

The accuracy rates of baseline models and our network on the ImageNet validation set are shown in Table 5. With the same backbone architecture, our model always obtains a higher accuracy rate compared with ResNet. For instance, The AFI-ResNet-152 increases the accuracy rate by 1.24%, decreases the number of parameters by 9.2%, and meanwhile decreases FLOPs by 10% compared with ResNet-152.

Additionally, the memory usage of our model is smaller than the base model(ResNet-50) in both training and testing process as shown in Table 6. In contrast with DenseNet-121 (Huang et al., 2017) which requires an enormous running space for keeping features of the total 24-layer in DenseNet block (3), our AFI-ResNet-50 maintains features at most 6 layers, therefore the memory usage of our model is much smaller. Besides, reduction of the output of convolution also helps to optimize memory costs during running time. Furthermore, our AFI-Module could release more running space via reducing intermediate gradients in memory with the method proposed by Pleiss et al. (2017).

5 Conclusion

In this paper, we proposed the AFI modules that can adaptively select features for feature reuse and improve the representational power of networks. The output feature of AFI blocks is aggregated by the channel-wise soft attention over a series of features. The structure of the AFI module is simple and can be used directly in existing state-of-the-art architectures easily. Experimental results show the effectiveness of AFI-Networks, which achieve competitive performance across multiple datasets. Compared with baseline models, AFI counterparts can achieve better performance with lower computational complexity. We believe that the AFI modules can be broadly applicable across various computer vision tasks, e.g., object detection, instance segmentation and semantic segmentation.

REFERENCES

Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *Advances in neural information processing systems*, pp. 4467–4475, 2017.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. 2018a.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018b.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2083–2090, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. *arXiv preprint arXiv:1911.06667*, 2019.

Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pp. 6389–6399, 2018a.

Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 254–269, 2018b.

Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 510–519, 2019.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 116–131, 2018.

Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 821–830, 2019.

Geoff Pleiss, Danlu Chen, Gao Huang, Tongcheng Li, Laurens van der Maaten, and Kilian Q Weinberger. Memory-efficient implementation of densenets. *arXiv preprint arXiv:1707.06990*, 2017.

Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7479–7489, 2019.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2017.

Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11534–11542, 2020.

Wenhai Wang, Xiang Li, Jian Yang, and Tong Lu. Mixed link networks. *arXiv preprint arXiv:1802.01808*, 2018a.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018b.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

Ke Zhang, Yurong Guo, Xinsheng Wang, Jinsha Yuan, Zhanyu Ma, and Zhenbing Zhao. Channel-wise and feature-points reweights densenet for image classification. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 410–414. IEEE, 2019.