IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 February 2023, accepted 31 March 2023, date of publication 6 April 2023, date of current version 11 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3265382

RESEARCH ARTICLE

Knowledge Distillation With Feature Self Attention

SIN-GU PARK AND DONG-JOONG KANG
Department of Mechanical Engineering, Pusan National University, Busan 43241, Republic of Korea
Corresponding author: Dong-Joong Kang (djkang@pusan.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) through the Korean Government (MSIT) under Grant 2021R1A2C1010057.

ABSTRACT With the rapid development of deep learning technology, the size and performance of the network continuously grow, making network compression essential for commercial applications. In this paper, we propose a Feature Self Attention (FSA) module that extracts correlation information between the hidden features of a network and a new method for distilling the correlation features to compress the model. FSA does not require a special module or network to match features between the teacher model and the student model. By removing the multi-head structure and the repeated self-attention blocks in the existing self-attention mechanism, it minimizes the addition of parameters. Based on ResNet-18, 34, the added parameters are only 2.00M and the training speed is also the fastest in comparison to benchmark models. It was demonstrated through experiments that the use of interrelationship loss between features can be beneficial for training student models, indicating the importance of considering correlation information in deep neural network compression. And it was verified through training from scratch on the vanilla without the pre-trained weight of the student model.

INDEX TERMS Knowledge distillation, self-attention, model compression, training from scratch.

I. INTRODUCTION

As deep learning technology rapidly develops in the field of computer vision, not only performance but also the size of network models continues to grow. Early simple Multi-Layer Perceptron (MLP) [1] was followed by convolutional neural networks (CNN) [2], which began to be called deep learning as the network model became deeper and broader to improve performance. In addition, a network called Vision Transformer (ViT) [3] was introduced, which combines a transformer [4], a method of natural language processing technology, into computer vision. Currently, large ViT-based network models that require large amounts of data boast the best performance (state-of-the-art) in several tasks.

Despite their performance, the amount of time and resources required to train such a giant network gradually became too much for individual researchers to experiment with or install on a typical device. This is contrary to industry's direction of high performance at a low cost, and research on network compression such as model pruning and network knowledge distillation [5] is being conducted to improve commercialization.

Unlike conventional model pruning algorithms, which directly reduce model size by removing less important neurons, network knowledge distillation improves accuracy by transferring knowledge from a large network to small one. This achieves performance that cannot be reached when trained only with a full student model. This method is an indirect model reduction method that enables the replacement of large models. For example, by handing over hidden layer features of the teacher model as hints to the student model, Fitnets [6] has shown similar performance to the teacher model. In addition, Yim et al. [7] proposed a method to extract and distill the relationship and flow information between hidden features of the teacher model.

There are still some limitations to these existing methods. First, when selecting hidden layers for use in knowledge distillation, flexibility is reduced due to a fixed number of

The associate editor coordinating the review of this manuscript and approving it for publication was Charalambos Poullis.

layers or location constraints such that feature sizes should be the same for comparison between features.

In this case, many combinations of feature selection are required to ensure that the number and location are appropriate to distill the teacher model's ability to extract features. Second, a module design for extracting knowledge to be distilled or searching for a matching method between two models is not simple; it requires an additional network of sizes comparable to student models. This makes it difficult to know whether the performance is due to the increasing the model capacity by the added network or the effect of knowledge distillation.

Thus, to solve these problems, we propose Feature Self Attention (FSA) module, which creates information that considers the interrelationships between hidden features. FSA consists of a feature embedding module (FEM) that embeds different sized hidden features extracted from a teacher model and a student model into the same token and a self-attention technique in a transformer. FSA can use all the hidden features of the model and refine features that take into account the interrelationships between hidden features, allowing it to distill richer information than conventional methods.

In addition, additional modules are not required for matching between multiple features of the teacher model and the student model. Features generated by FSA are validated for the classification through a simple fully connected layer (FSA_FC) classifier, and then distilled into the student model.

The main contributions of this paper are summarized as follows.

1. We propose FSA, a new model that refines and distills interrelationship information by using all hierarchical features information of the teacher model. Through this, it is not necessary to design a module that calculates efficient matching with the student model feature.

2. We find that the student model performance is better to train the interrelationship information between hidden layers first than to train with two losses including task loss function simultaneously. This shows that the interrelationship between the features is effective for knowledge distillation.

3. In the FSA module that extracts interrelationship information, repetitive stacking of the self-attention mechanism was excluded, and additional parameters for knowledge distillation were minimized with a simple network design.

II. RELATED WORKS

Knowledge distillation is generally a method to help training by transferring information of teacher models that are large in size and have high performance to relatively small student models, and to obtain performance beyond the limits of existing student model capacity. There are three main categories depending on what you treat as the network knowledge to distill.

A. RESPONSE-BASED DISTILLATION

First, response-based distillation is a method of dealing with knowledge that distills the output value of the teacher model. It was introduced in the early stages. Since the model output value or the last layer is mainly refined and used, this method can be used for various tasks such as object detection, human pose judgment, and heat map division. Influential studies in these fields used the soft targets technique [5], [8], which uses the output probability distribution of teacher model as labels for student model. This method is often used with other categories of knowledge distillation methods.

B. FEATURE-BASED DISTILLATION

The second category is feature-based distillation: a knowledge distillation method that learns to create similar features by matching between specific layers of a teacher model and a student model. Led by the Hinting method of teacher model feature learning [6], Zagoruyko and Komodakis [9] created an attention map with a specific feature of the teacher model and distilled it to the student model. Furthermore, Chen et al. [10] proposed a method that calculates the interaction between the features of the teacher model at multiple layers and the features of the student model, and uses the resulting values as weights in the matching combination. These methods require time consuming experiments or specific modules on how to match and find a location to extract features from each model.

C. RELATION-BASED DISTILLATION

In relationship-based distillation, features of two specific layers in the teacher model are measured for similarity and relationship as a similarity evaluation function, and their values are distilled [11], [12].

Alternatively, methods of distillation to use the flow of information between two specific layers [13] are being studied. It refines network as a graph to distill the flow of information. The edge of the graph represents the relationship between features and this method requires a module or network for the graph. Figure 1 briefly shows two structures comparison of the proposed model and the basic model of feature based knowledge distillation.

III. METHODOLOGY

First, Section III-A to III-C briefly explains the backbone models, their selection reasons, and mechanisms of modules used in the proposed method. Detailed explanations of learning hidden feature interrelationships are then given in Sections III-D to III-G.

A. BACKBONE MODEL

To be a suitable backbone model, it is preferable that the network can be stacked in small network blocks or divided into several stages. That is, it is preferred that there are regions in which the network configuration changes or the feature size changes. This is because FSA is proper for using various hierarchical features in such region.

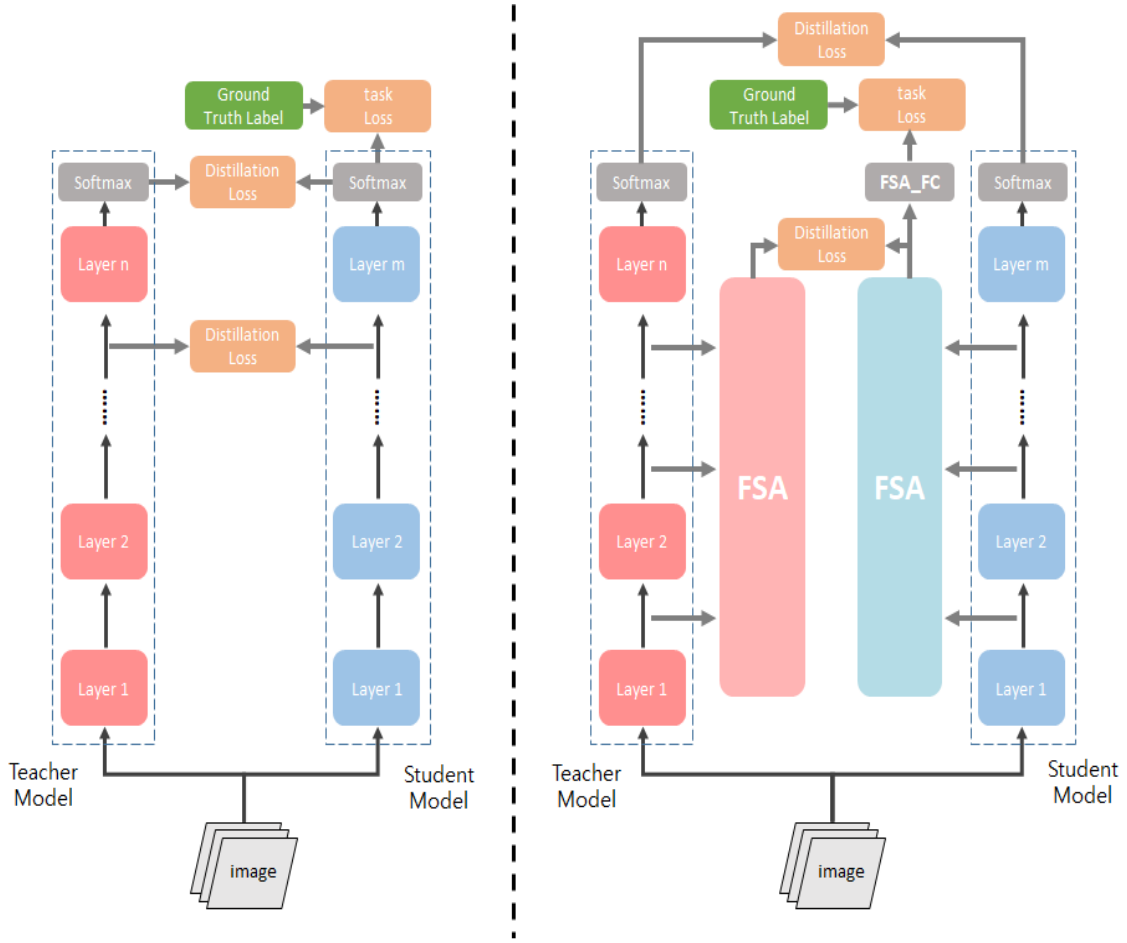

FIGURE 1. Basic knowledge distillation model(Left). Proposed model(Right). The output value of student model was used for the classification in the basic model. The proposed model refines the features of the FSA through a simple fully connected layer classifier (FSA_FC) and uses them for the classification.

ResNet [14] and Swin [15] are suitable models for this; however, Swin still lacked benchmark data used as the backbone model for knowledge distillation. ViT [3], similar to Swin and often used as a backbone model, cannot extract hierarchical features because the stacked blocks have the same input/output feature size. Thus, the ResNet model was selected as the backbone model.

B. ResNet

Overstacking convolutional layers to increase the prediction accuracy in CNN-based networks such as AlexNet [16] and VGGNet [17] rather could reduce the performance of the model due to the well-known gradient vanishing problem. In the ILSVRC [18] and COCO [19] conferences, He et al. [14] proposed residual blocks, as shown in Figure 2, to address this problem and show no gradient loss even in deep models of 152 layers.

ResNet has some different configurations of residual blocks as well as deep structure of 152 layers, so there are different layers of models, as shown in Table 1. In this paper, Resnet-18 was used as a student model and other models were used as a teacher model. Knowledge distillation was tested by extracting hidden features in four stages based on the layer parts where the composition of the residual block changes.

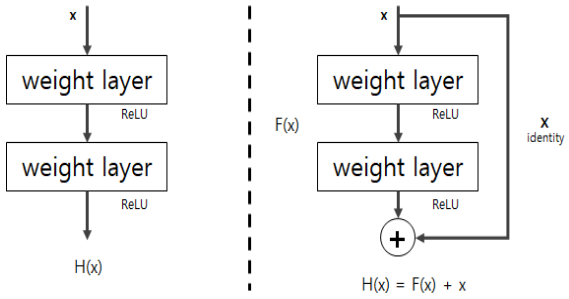

FIGURE 2. Basic convolution block (left), residual block (right).

C. SELF-ATTENTION

In the field of natural language processing, the transformer model creates an embedding feature of the same size, called token, and then generates other tokens by adding

TABLE 1. Convolutional block configurations and stacking counts for each model stage presented in ResNet [14].

Stage name	Output size	ResNet-18	ResNet-34	ResNet-50	ResNet-152
stage 1	56 × 56	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix}\times 2$	$\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix}\times 3$	$\begin{bmatrix} 1\times1, & 64 \\ 3\times3, & 64 \\ 1\times1, & 256 \end{bmatrix}\times 3$	$\begin{bmatrix} 1\times1, & 64 \\ 3\times3, & 64 \\ 1\times1, & 256 \end{bmatrix}\times 3$
stage 2	28 × 28	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix}\times 2$	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix}\times 4$	$\begin{bmatrix} 1\times1, & 128 \\ 3\times3, & 128 \\ 1\times1, & 512 \end{bmatrix}\times 4$	$\begin{bmatrix} 1\times1, & 128 \\ 3\times3, & 128 \\ 1\times1, & 512 \end{bmatrix}\times 8$
stage 3	14 × 14	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix}\times 2$	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix}\times 6$	$\begin{bmatrix} 1\times1, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \end{bmatrix}\times 6$	$\begin{bmatrix} 1\times1, & 256 \\ 3\times3, & 256 \\ 1\times1, & 1024 \end{bmatrix}\times 36$
stage 4	7 × 7	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix}\times 2$	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix}\times 3$	$\begin{bmatrix} 1\times1, & 512 \\ 3\times3, & 512 \\ 1\times1, & 2048 \end{bmatrix}\times 3$	$\begin{bmatrix} 1\times1, & 512 \\ 3\times3, & 512 \\ 1\times1, & 2048 \end{bmatrix}\times 3$

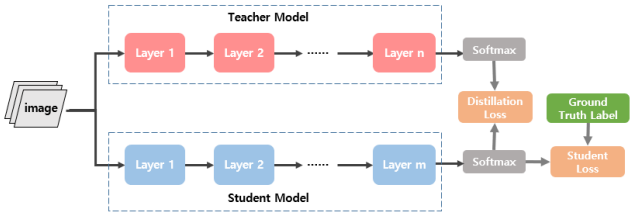

FIGURE 3. Traditional response-based knowledge distillation model.

information considering the context in the sentence using Self-Attention (SA).

The transformer model is trained by stacking modules including SA, and shows state-of-the-art performance in most tasks in the field of natural language processing.

The process of SA is expressed by the following equation.

$$Attention\,(\mathrm{Q},\mathrm{K},\mathrm{V}) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{1}$$

$$\mathrm{Q} = \mathrm{T}W_q \in \mathbb{R}^{N\times d}, \tag{2}$$

$$\mathrm{K} = \mathrm{T}W_k \in \mathbb{R}^{N\times d}, \tag{3}$$

$$\mathrm{V} = \mathrm{T}W_v \in \mathbb{R}^{N\times d}, \tag{4}$$

Q, K, and V are the Query, Key, and Value, and d is the dimension of weights.

When the number of words in a sentence is N and the i-th word token is t_i,

$$\mathrm{T} = \{t_1, t_2, \cdots, t_N\} \in \mathbb{R}^{N\times E}. \tag{5}$$

where E is an embedding dimension.

Dosovitskiy et al. [3] proposed a method of embedding image data like tokens in natural language processing and used them in the transformer model, achieving high performance in various tasks of computer vision. Since then, various methods of embedding image data like tokens of natural language processing have been studied and combined with the transformer model. However, according to the demand for performance increase, the transformer module is stacking wider and deeper, and the size of the model is becoming too large. In the proposed approach, we show that SA is effective even if it is used only once without stacking.

D. FSA

To explain the difference from the existing model, the structure of simple knowledge distillation is shown in Figure 3.

Existing loss functions are mainly composed of the following distillation loss and task loss. The distillation loss function allows the student model to imitate the teacher model so that it can learn quickly. The task loss function preserves the generalization performance when only the student model is separated and used after knowledge distillation is completed.

In contrast, the FSA network consists of FEM modules, SA modules, and a fully connected (FC) layer classifier as well as a teacher model and a student model that outputs classification probability distribution results. FEM modules convert the extracted hidden features into embedded features of the same size, SA modules add interrelationship information to the features, and the FC layer classifier performs the classification task. The structure of the entire network is shown in Figure 4.

When the hidden layers of the teacher model and the student model are matched and distilled, the layers of the student model are trained to mimic only the feature extraction ability of the matched teacher model's layers. In our method, information about how the selected hidden layers interact and communicate is added to the feature by the FSA. Also it is distilled using all hierarchical level's features of teacher model.

The interrelationship between features is distilled through FSA while the overall feature extraction ability and classification performance of the model is distilled through the output value of the teacher model. After distilling only the FSA features in the early stages of training, adding the classification ability in next stage of training showed higher performance than distilling both from the beginning.

For the task loss function, we used classification probability distribution obtained by a simple FSA_FC for features

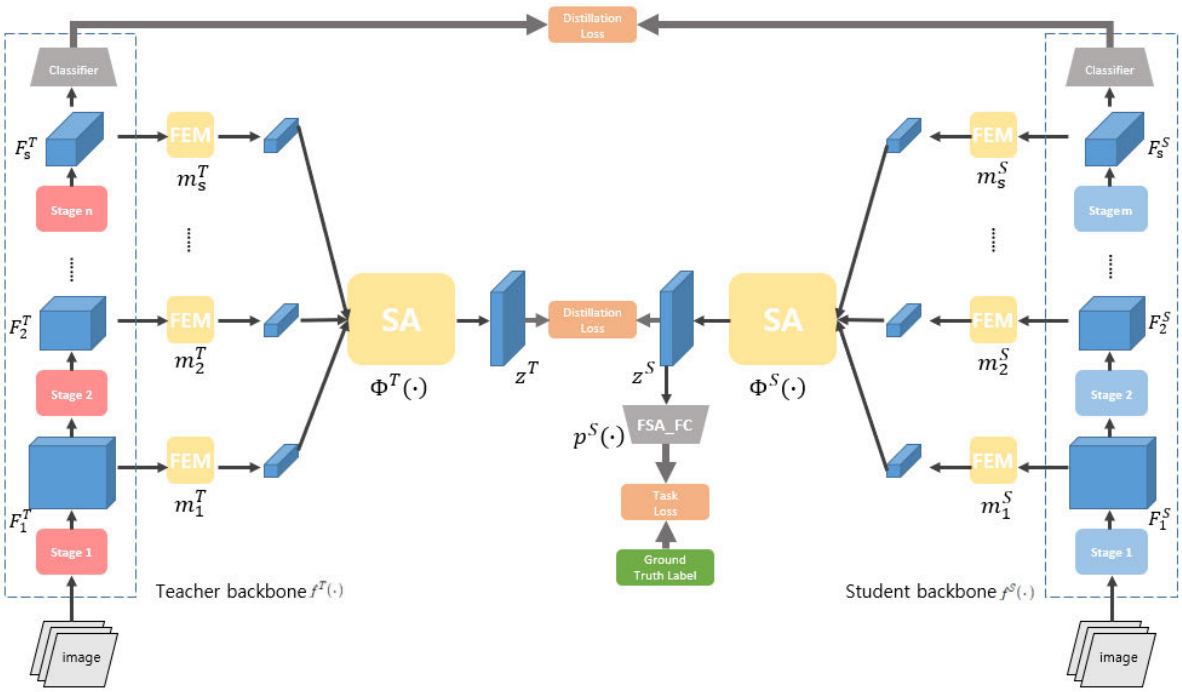

FIGURE 4. FSA network structure. The FSA_FC of the teacher model has been removed after training the teacher model. The FSA_FC of the teacher model and the student model is the same type of module but has different weights.

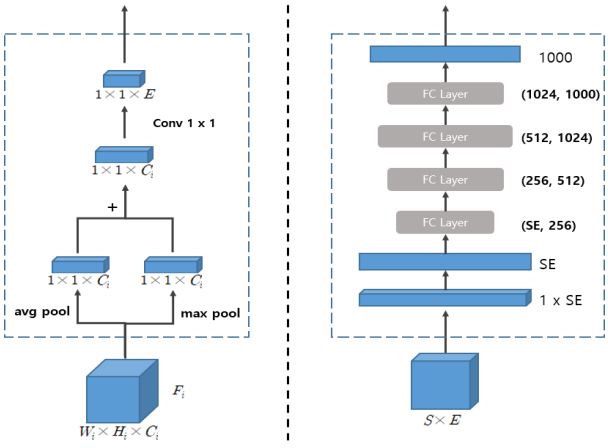

FIGURE 5. FEM (left), FSA_FC (right).

extracted with FSA, rather than the classification probability distribution at the top of the student model. This is to prevent training with the capacity of the student model, not knowledge distillation, by calculating the loss function comparing the output value of the student model and the ground truth label.

The FSA of the teacher model was trained by comparing the FSA_FC output with the ground truth label. Similarly, comparing the FSA_FC output of the student model with the ground truth label was also effective in improving the FSA performance of the student model.

E. FEM

There have been several studies on how to embed images or features. In most cases, additional modules or networks were required and the size of whole network become bigger. To avoid this, many studies, such as ViT and Swin, used methods to divide input image data or hidden features into n × n regions and refine and embed each region. This paper suggests an FEM that minimizes the size of the embedding module without an additional refinement process, thus reduces the overall model size even when used for each hidden feature. FEM is responsible for embedding different sized hierarchical hidden features output from each stage of the teacher model or student model into tokens of the same size that can be used in SA. As shown in Figure 5, first, hidden features are independently processed from average pooling and maximum pooling. The pooling uses $W_i \times H_i$ sized kernel instead of a typical 3 × 3 size to create channel-wise features. After pooling, two features are combined for embedding. Then, through 1 × 1 convolution, the embedding is completed by unifying the different channels C_i for each hidden feature into the embedding dimension E.

The 1 × 1 convolution not only changes the number of channels, but also has a channel-wise attention effect as an operation using all channels. FEM is designed with only two pooling and one convolutional layer, rather than an additional network for embedding, minimizing the size increase of the entire model.

F. FSA_FC

HSAKD [15] performed a task of classifying which rotational transformations have been given to the input data through an additional network using a hidden feature. Then, it distills the added network's output to the student model.

In a typical learning model, it is difficult to infer what the hidden feature means unless it is a feature extracted from the end layer. However, if the added task is performed by the hidden features, regardless of the task's difficulty, it could guide a meaningful direction in feature extraction, thereby can distill high-quality features. Inspired by this, the proposed model is designed to use a simple classifier that performs same classification at the end layer. This classifier uses features generated from the FSA. Unlike HSAKD, this paper distills not only the output of the added classifier but also feature generated from the FSA. The simple classifier added to the end of the FSA uses only four fully connected layers, as shown in right side of Figure 5, and belongs to both the teacher model and the student model. The FSA_FC of each side model is used only for training the corresponding model FSA, and is not used for distillation learning in the counterpart model.

G. TRAINING TEACHER MODEL

The network shown in Figure 6 is used to train the teacher model.

F^T refers to the backbone model of the teacher model, and $F^{T_i} \in \mathbb{R}^{N\times W_i\times H_i\times C_i}$ refers to the feature of the i-th stage of the teacher model. In this case, W is the width of the feature, H is the height of the feature, and C is the number of channels of the feature.

$m^{T_i}\left(F^{T_i}\right) \in \mathbb{R}^{N\times S^T\times E}$ stands for the i-th FEM, where S is the number of stages of the backbone model, and E is the number of embedded channels.

Φ^T refers to SA and is a module that maintains an input/output size.

z^T refers to a feature extracted by FSA and may be expressed by the following equation.

$$z^T = \Phi(\{m^{T_1}(F^{T_1}), m^{T_2}(F^{T_2}), \cdots, m^{T_S}(F^{T_S}),\}), \tag{6}$$

$$z^T \in \mathbb{R}^{N\times S^T\times E}. \tag{7}$$

$p^T(\cdot)$ is a classifier consisting of a fully connected layer and outputs a probability distribution that receives a feature generated by FSA and predicts a label of the input data to obtain a Cross Entropy loss function with the label of the input data.

The loss function used for teacher model training is as follows

$$L^{T_{FSA}} = L_{CE}\left(p\left(z^T(x)\right), y\right). \tag{8}$$

The cross entropy loss function used here is as follows

$$L_{CE}(x, y) = -\sum(x_i \log_2 y_i). \tag{9}$$

In x ∈ X and y ∈ X, X and Y is a training image data set and a label data set, respectively.

The backbone teacher model freezing weights are pre-trained with ResNet 34, 50, 152 [14] so that only the FSA and FSA_FC parts were trained.

H. TRAINING STUDENT MODEL

The network used to train the student models is shown in Figure 4. f^S refers to the backbone model of the student model, and $F^{S_i} \in \mathbb{R}^{N\times W_i\times H_i\times C_i}$ refers to the feature of the i-th stage of the student model. $m^{S_i}\left(F^{S_i}\right) \in \mathbb{R}^{N\times S^S\times E}$ refers to the i-th FEM. The superscript S means that it is a component of the student model and exponential S is the number of stages of the backbone model. Φ^S refers to a feature extracted as SA, and z^S refers to a feature extracted as FSA. $p^S(\cdot)$ outputs a probability distribution that predicts a label of input data by receiving a feature generated by FSA with a fully connected layer classifier.

In up to 20 epochs out of a total of 100 epochs, the output values of the two backbone models are not used, and the loss function is obtained only with features generated by FSA to only train the interrelationship between the hidden features. The loss function uses the mean square error. We then further distill the output probability distribution of the two backbone models up to 50 epochs into the KL-Divergence loss function, which calculates the difference between the probability distributions. We added a cross entropy loss to the total loss to train the generalization ability of the student backbone model. The cross entropy loss compares features of the FSA_FC with label of input data.

The loss function used in training the student model is as follows

$$L_{FSA\_f} = L_{MSE}\left(z^T(x), z^S(x)\right), \tag{10}$$

$$L_{KL} = D_{KL}\left(f^T(x)\,||\,f^S(x)\right), \tag{11}$$

$$L_{FSA\_FC} = L_{CE}\left(p\left(z^S(x)\right), y\right), \tag{12}$$

$$L = \lambda_1 L_{FSA\_f} + \lambda_2 L_{KL} + \lambda_3 L_{FSA\_FC}, \tag{13}$$

$$\left[\lambda_1, \lambda_2, \lambda_3\right] = \begin{cases} [1\ 0\ 0] & (0 \le epoch < 20) \\ [1\ 1\ 0] & (20 \le epoch < 50) \\ [1\ 1\ 1] & (50 \le epoch < 100) \end{cases}. \tag{14}$$

The mean square error loss function and the KL-Divergence loss function used here are as follows:

$$L_{MSE}(x, y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2, \tag{15}$$

$$D_{KL}(P\,||\,Q) = \sum_{x\in X} P(x)\log\left(\frac{P(x)}{Q(x)}\right). \tag{16}$$

IV. EXPERIMENTS

The dataset used ImageNet-1K [16] consisting of approximately 1.28 million training data with 1,000 classes and 50,000 validation data. An image was randomly cropped in 224 × 224 size and trained using only random horizontal inversion data multiplication set to p = 0.5. The teacher model was tested with three types, ResNet-34, ResNet-50 and

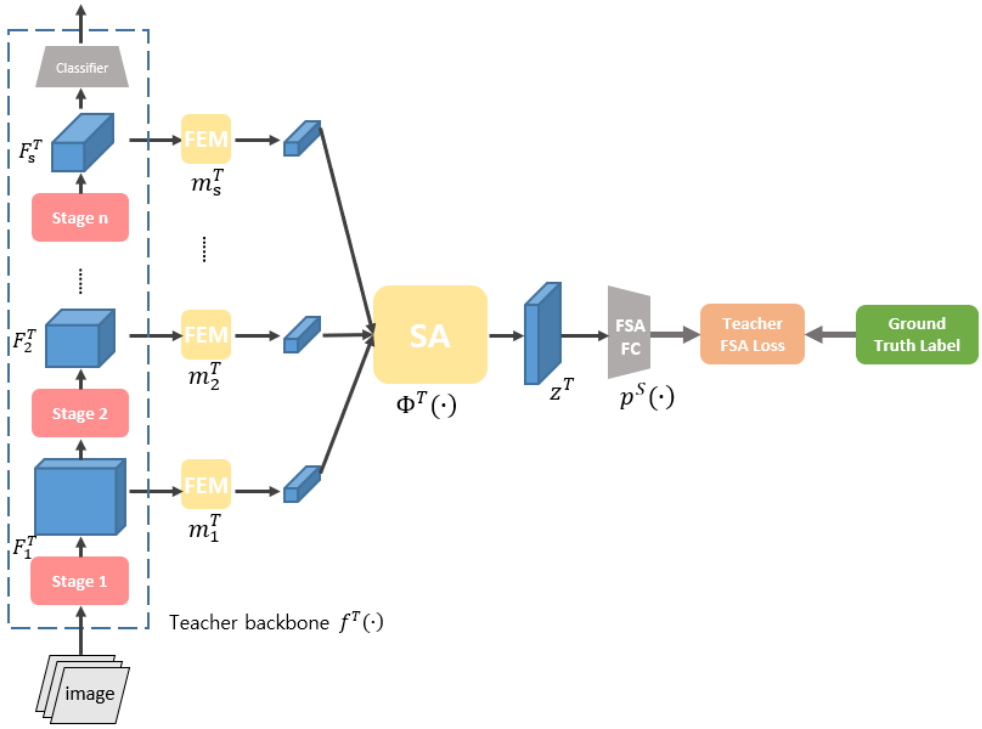

FIGURE 6. Network for teacher model.

ResNet-152, and the student model was fixed to be trained with ResNet-18.

Train was conducted by a total of 100 epochs, with a batch size of 128, optimizer using Stochastic Gradient Descent, and the initial learning rate was set to 0.01 when the teacher model used ResNet-34, 0.005 with ResNet-50 and 0.0001 with ResNet-152. The learning rate was reduced 1/10 by every 30, 60, and 90 epochs.

V. RESULTS

A. FEATURE REFINING PERFORMANCE OF FSA

When training the teacher model FSA, the main task was performed after passing through a simple FSA_FC with features containing the hidden features of the teacher backbone model. If the information of the backbone teacher model was well refined as intended, it would have similar performance to the backbone teacher model; however, if a meaningless feature is generated contrary to the intention, it would be difficult to produce meaningful performance only with FSA_FC. Therefore, to evaluate how well the FSA has refined the information of the teacher model, a performance comparison of the FSA_FC of several teacher models with the backbone teacher model was conducted.

Experimental results are shown in Table 2. The features generated with FSA used a simple FSA_FC, but similarities can be seen when comparing the performance of the learned open-source backbone teacher model under optimal conditions.

Since there is a slight difference of about 2%p from Top-1 and 1-1.5%p from Top 5, it can be seen that the information of the teacher model to be distilled into the student model is well refined.

B. COMPARISON OF KNOWLEDGE DISTILLATION PERFORMANCE

Table 3 shows the ImageNet-1K classification performance of the student model and the performance of the FSA_FC after distillation learning with features including the interrelationship between the hidden features of other teacher models. Table 4 shows the total number of parameters added for this.

Student models that completed distillation learning from ResNet-34 and ResNet-50 teacher models improved their student models by Top-1: 1.55%p/1.65%p, and Top-5: 1.77%p/1.96%p, respectively, compared to vanilla models (Top-1: 66.11% and Top-5: 86.90%, respectively). However, the student model, which completed distillation learning from the ResNet-152 teacher model, performed worse at -8.73%p and -5.05%. This is presumed to be because, given the number of block stacks per each model stage shown in Table 1, the significantly increased number of stage 3 blocks in ResNet-152 broke the balance of the correlative roles that the stage has in the entire model.

TABLE 2. Comparison of performance of the backbone teacher model with performance of FSA_FC.

Model	backbone output		FSA_FC output	
	Top-1	Top-5	Top-1	Top-5
ResNet-34	73.31	91.41	71.44	90.14
ResNet-50	76.81	92.93	74.46	91.87
ResNet-152	78.31	94.04	76.68	93.01

TABLE 3. Comparison of knowledge distillation performance according to teacher model.

Teacher	Student	backbone output		FSA_FC output	
		Top-1	Top-5	Top-1	Top-5
ResNet-34	ResNet-18	**67.66**	**88.67**	63.42	85.28
ResNet-50		**67.76**	**88.86**	64.66	85.72
ResNet-152		57.38	81.85	56.97	80.6
-	ResNet-18	66.11	86.9		

TABLE 4. Comparison of the number of parameters before and after adding the FSA network.

Model	ResNet-18	ResNet-34	ResNet-50	ResNet-152
backbone	11.69M	21.8M	25.56M	60.19M
backbone + FSA	13.69M	23.8M	27.93M	62.56M
adding params.	+ 2.00M	+ 2.00M	+ 2.37M	+ 2.37M

C. PERFORMANCE COMPARISON BY PRECEDENCE OF FSA FEATURE DISTILLATION

Table 5 compares the performance of two cases: (1) when output loss was step-wisely added after pretrained with only the FSA loss and (2) when the FSA loss and output loss were simultaneously trained from the beginning. In Eq. (13), the former coefficients of each loss term follow Eq. (14) and the latter coefficients follow Eq. (17)

$$[\lambda_1, \lambda_2, \lambda_3] = \begin{cases} [1\ 1\ 0] & (0 \le epoch < 50) \\ [1\ 1\ 1] & (50 \le epoch < 100)\,. \end{cases} \tag{17}$$

When output loss is further distilled after pretrained of the interrelationship information, the performance improvement was slightly higher with Top-1:0.83%p/0.50%p and Top-5:1.11%p/0.34% when the teacher model is ResNet-34 and ResNet-50, respectively.

TABLE 5. Performance comparison according to the combination of initial loss during distillation.

Teacher	Student	FSA loss		FSA loss + output loss	
		Top-1	Top-5	Top-1	Top-5
ResNet-34	ResNet-18	**67.66**	**88.67**	66.83	87.56
ResNet-50		**67.76**	**88.86**	67.26	88.52
-	ResNet-18	66.11	86.9		

TABLE 6. Teacher Model: ResNet-34. Student model: ResNet-18. ImageNet performance improvement benchmark in knowledge distillation. KD: knowledge distillation [20], AT: attention transfer [9], FT: factor transfer [21], CRD: contrastive representation distillation [22], and Tf-KD: teacher-free knowledge distillation [23].

	Acc diff	Epochs	Training Time
KD[20]	+ 1.48	100	60hr 04min
AT[9]	+ 1.15	100	59hr 07min
FT[21]	**+ 1.81**	91	55hr 06min
CRD[22]	+ 1.06	100	356hr 31min
Tf-KD[23]	+ 0.77	90	46hr 34min
FSA (ours)	+ 1.55	100	**40hr 13min**

D. PERFORMANCE COMPARISON FOR BENCHMARK MODELS

Table 6 compares the performance improvements achieved after distilling knowledge into student model ResNet-18 using teacher model ResNet-34 with ImageNet benchmarks.

The method proposed in this paper does not provide the greatest improvement; however, our method shows significant performance in training time.

VI. CONCLUSION

In this paper, we proposed FSA, which extracts interrelationship information between hidden features by utilizing all the hierarchical features of the model.

This does not require a special module or network to match the knowledge to be distilled between the teacher model and the student model. When only the hidden feature relational information was distilled from the beginning, it was found that performance improved more than when simultaneously training the main task from the beginning. Accordingly, we verified that information on the role of each layer in the model helps the layer to learn the feature extraction ability.

In future knowledge distillation research, this may be a method to replace feature matching modules or matching networks.

In addition, although it is trained from scratch on the vanilla model, FSA achieves performance improvements comparable to the open-source distillation model trained under optimal conditions. It is possible to reduce the constraint on the existence of pre-trained weights in the process of selecting the knowledge distillation student model.

REFERENCES

[1] H. Ramchoun, M. Amine, M. A. J. Idrissi, Y. Ghanou, and M. Ettaouil, "Multilayer perceptron: Architecture optimization and training," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, pp. 26–30, Jan. 2016.
[2] K. Teilo, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.
[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–22.
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[5] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[6] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," in *Proc. ICLR*, 2015, pp. 1–13.
[7] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4133–4141.
[8] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. NeurIPS*, 2014, pp. 1–9.
[9] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. ICLR*, 2017, pp. 1–13.
[10] D. Chen, J. P. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, and C. Chen, "Cross-layer distillation with semantic calibration," in *Proc. AAAI*, 2021.
[11] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proc. CVPR*, Jun. 2019, pp. 3967–3976.
[12] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *Proc. SIGKDD*, 2017, pp. 1285–1294.
[13] N. Passalis, M. Tzelepi, and A. Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *Proc. CVPR*, Jun. 2020, pp. 2339–2348.
[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[15] C. Yang, Z. An, L. Cai, and Y. Xu, "Hierarchical self-supervised augmented knowledge distillation," in *Proc. IJCAI*, 2021, pp. 1–13.
[16] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012.
[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," 2014, *arXiv:1409.0575*.
[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.
[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. Deep Learn. Represent. Learn. Workshop (NIPS)*, 2014, pp. 1–9.
[21] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2760–2769.
[22] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *Proc. 8th Int. Conf. Learn. Represent.*, 2020, pp. 1–19.
[23] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, "Revisiting knowledge distillation via label smoothing regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 3903–3911.

SIN-GU PARK received the B.S. degree from the School of Mechanical Engineering, Pusan National University, Busan, South Korea, in 2020, where he is currently pursuing the master's degree. His research interests include deep learning, machine learning, pattern recognition, and network compression.

DONG-JOONG KANG received the B.S. degree in precision engineering from Pusan National University, Busan, South Korea, in 1988, and the M.S. and Ph.D. degrees in mechanical and automation design engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1990 and 1999, respectively. From 1997 to 2000, he was a Research Engineer with the Samsung Advanced Institute of Technology (SAIT). He is currently a Professor with the School of Mechanical Engineering, Pusan National University. His current research interests include machine vision, machine learning, and visual inspection in factories. He was an Associate Editor of the *International Journal of Control, Automation, and Systems*, from 2007 to 2019.