NeuRes: Highly Activated Neurons Responses Transfer via Distilling Sparse Activation Maps

SHARMEN AKHTER, MD IMTIAZ HOSSAIN, MD DELOWAR HOSSAIN, AND EUI-NAM HUH, (Member, IEEE)
Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, South Korea
Corresponding author: Eui-Nam Huh (johnhuh@khu.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068; and in part by the IITP grant funded by the Korea Government (MSIT), Artificial Intelligence Convergence Innovation Human Resources Development, Kyung Hee University, under Grant RS-2022-00155911.

ABSTRACT In recent years, Knowledge Distillation has obtained a significant interest in mobile, edge, and IoT devices due to its ability to transfer knowledge from the large and complex teacher to the lightweight student network. Intuitively, Knowledge Distillation refers to forcing the student to mimic the teacher’s neuron responses to improve the generalization of the student by deploying the distillation losses as the regularization terms. However, the non-linearity of the hidden layers and the high dimensionality of the feature maps make the knowledge transfer a rigorous task. Though numerous methods have been proposed to transfer the teacher’s neuron responses in the form of diverse feature characteristics such as attention, contrastive representation, and so on, to the best of our knowledge, no prior works considered feature-level non-linearity during distillation. In this work, we ask, does feature-level non-linearity-based approaches improve student performance? For investigating those concerns, we propose a novel knowledge distillation technique called the NeuRes (Neuron’s Responses) via distilling the Sparse Activation Maps (SAMs) to transfer the highly activated Neurons Responses to the student to enhance the representation capability. Proposed NeuRes selects the highly activated neuron responses that produce Sparse Activation Maps (SAMs) while transferring the knowledge based on activation normalization. Our proposed NeuRes also transfers the translation invariant features using auxiliary classifiers and augmented data to improve students’ generalization. The detailed ablation studies and extensive experiments on model compression, transferability, adversarial robustness, and few-shot learning verify that NeuRes outperforms state-of-the-art distillation techniques on the standard benchmark datasets.

INDEX TERMS SAMs, NeuRes, knowledge distillation, representation learning, knowledge transfer, response distribution, node response, activation boundary shifting, activation boundary scaling.

I. INTRODUCTION
Since few decades, vision-based deep learning approaches have attained immense attention by providing great performances in diverse visual-based tasks such as segmentation, detection, classification, recognition, reconstruction, and so on [1], [2], [3], [4], [5], [6], [7], [19]. However, the heavy and complex deep architecture demands high computational resources and energy consumption. This fact makes the large and complex deep architectures unsuitable to deploy at low computational resource-based and real-time applications such as mobile, IoT, and edge computing devices [20]. Moreover, 6G networks and AI-enabled edge, cloud, and fog computing-based infrastructures demand fast and rational learning systems i.e., lightweight AI experts, to facilitate real-time execution in low computational terminal devices such as smartphones, IoT, vehicular edge devices and so on [51], [52]. On the other hand, lightweight architectures perform
relatively poor than heavy and complex architectures due to having lower generalization capability [8], [9]. To this end, numerous approaches have been suggested to compress [12], [13], [15], prune [10], [11], quantize [14] or mimic [16], [17], [22], [23] teacher networks into or by a small and lightweight student architecture. Knowledge Distillation is a very effective technique that provides explicit supervision to the lightweight student to achieve the teacher’s performance without increasing the number of parameters and computational complexity [16], [18], [22], [23].

Hinton et al. [23] first introduced the knowledge distillation (KD) technique to transfer predictive logits distributions to the student from the teacher(s). KD [23] transfers predictive logits-level knowledge via softening the logits distribution by dividing by a constant factor called temperature where no distillation loss is deployed into intermediate feature space. Due to having lower representational capacity than the teacher networks, the students struggle to mimic the intermediate feature representations i.e., latent space [24], [25]. Numerous studies show that reducing only Kullback-Leibler (KL) divergence between the global logits distributions of the student and teacher is not enough. To improve this scenario, numerous feature-level knowledge distillation techniques have been proposed to force the student to mimic the characteristics of the intermediate features of the teacher such as contrastive representations, attention, instance similarity, inter-channel co-relation, and so on [21], [26], [27], [28]. Intuitively, the principle goal of the knowledge distillation approach is to improve student generalization by forcing them to learn the neuron responses of the teacher network(s). The feature-level knowledge distillation techniques perform better compared to the logits-level distillation techniques. Though transferring this intermediate representational knowledge improves the student’s generalization capability, due to the nonlinearity of the intermediate activation maps, achieving optimal performance is still a rigorous task.
The ReLU function is defined by \( \sigma(x) = \max(0, x) \). Fig. 1(a) and Fig. 1(b) show the conventional ReLU activation function and the process of coordinate shifting, respectively.

The activation functions improve the non-linearity. The neural networks with the mixer of activation boundaries estimate a high-performing function \([39], [40]\). A combination of the activation boundaries creates a better decision boundary of a neural network \([41]\). These properties of activation functions indicate that transferring knowledge through refinement via an activation function improves performance by creating better decisions. Heo et al., \([33]\) showed that transferring knowledge via creating an activation boundary improves the performance in the knowledge transfer task. However, \textit{does that transferring activation boundary provide optimal performances in knowledge distillation tasks?} Fig. 3 shows that though transferring the intermediate activation maps provides improved performances, they are sub-optimal.

In this work, we focus on the question, \textit{does that high dimensional and dense representations provide optimal performances? Are transferring this dense knowledge highly error-sensitive to input data?} Stanton et al., \([25]\) discussed that though the students struggle to learn the teacher's knowledge perfectly, they may have the capacity to learn better compared to the existing outcomes. Liu et al., \([28]\) proved that directly transferring the activation maps to the student i.e., pixel-to-pixel knowledge transfer is very sensitive to errors and less effective \([30], [33]\).

Inspired by these observations, we focus on transferring highly activated neuron responses via distilling sparse activation maps and propose NeuRes by shifting and scaling the activation boundary. Fig. 1 shows the overview of the process of shifting and scaling the activation boundary of the ReLU activation function where the black and orange colored axis denote the original and shifted coordinate systems, respectively. ReLU refers to the rectified linear unit.

In the deep learning model, ReLU provides non-linearity. The ReLU function is defined by \( \sigma(x) = \max(0, x) \). Fig. 1(a) and Fig. 1(b) show the conventional ReLU activation function and the process of coordinate shifting, respectively.

The activation maps are normalized to the range of \([-1, 1]\). Fig. 1(c) visualizes the process of highly activated neuron response selection based on the transformed coordinates. The deactivated responses in the new coordinate systems are avoided. In Fig. 1(d), the activation boundary is scaled by a hyperparameter, called \( \xi \).

The proposed NeuRes transfers the highly activated neurons to the student which aims to mimic the sparse and highly activated neuron responses of the teacher network. The sparse knowledge-transferring technique solves the problem of high dimensionality and provides implicit non-linearity that makes the student improve generalization by utilizing their own capacity. As the student has lower generalization capability and lower activated neuron responses are dropped during training the student, we also feed the augmented data to both teacher and student that helps the student learn the teacher’s translation invariant features on augmented data for improving generalization capability (see Experiments sections (IV) for details). In Fig. 2, we show the overview of the process of feeding both augmented and original data to help the student improve generalization using contrastive knowledge. Fig. 2(a) and Fig. 2(d) indicates the highly activated neuron responses of the teacher and student networks, respectively. Fig. 2(c) and Fig. 2(d) show the contrastive knowledge between the outcomes of the auxiliary classifiers for original and augmented data. The contrastive knowledge with other instances is also deployed as the regularization term.

Let’s consider, \( A_i^T \) and \( A_i^T \) denote the \( i \)-th activation maps of the student \( \mathcal{X}^\text{S} \) and the teacher \( \mathcal{X}^\text{T} \), respectively. Firstly, we filter the highly activated neuron’s responses to obtain the Sparse Activation Maps (SAMs), \( S_i^T \) and \( S_i^S \) from \( A_i^T \) and \( A_i^S \), respectively, based on the magnitude by shifting and scaling the activation boundary depicted in Fig. 1. The highly activated and sparse neuron’s responses are then transferred to the student \( \mathcal{X}^\text{S} \) from the teacher \( \mathcal{X}^\text{T} \) along with the contrastive knowledge over input instances and augmented data shown in Fig. 2. The teacher predictive logits-level knowledge (logits distribution) is improved by reducing the KL divergence among the auxiliary (that is trained using augmented data and self-supervision approach)
and the main classifiers [38]. During transferring the knowledge, the student is then forced to learn the improved predictive logits-level knowledge of the teacher by deploying the distillation losses. Fig. 2 shows the overall process of distilling teacher knowledge. During distillation, we consider four losses, such as \( L_{kd}, L_{ce}, L_{ax-st} \), and \( L_{res} \). The overall loss function \( (L_{NeuRes}) \) is shown in Eq. 12. The details about the proposed technique are discussed in section III. Our core contributions can be summarized as:

- We provide an extensive investigation on the hypothesis that activation boundary shifting and scaling is a very effective technique to make the student learn better along with data augmentation and self-supervision approach.
- We show that transferring sparse representations is better compared to transferring dense representations along with data augmentation.
- Based on the analysis and observation, we propose a novel technique called NeuRes to transfer highly activated neuron responses and learning translation invariant features by the student via distilling Sparse Activation Maps (SAMs).
- We provide extensive results on popular benchmark standards, CIFAR100, STL10, and TinyImageNet for image classification, transferability, adversarial robustness transfer, and few-shot training tasks with detailed ablation studies.

II. RELATED WORKS
A. FEATURE-LEVEL AND LOGITS-LEVEL KNOWLEDGE DISTILLATION
Knowledge distillation was introduced by Hinton et al. in [23]. The core concept was to reduce KL divergence among the logits between the teacher and students. To soften the logits distribution that makes the student easy to learn they divide the logits distribution by a constant factor called temperature, \( \tau \). This technique basically transfers predictive logits-level knowledge i.e., predictive distribution to the student. That kind of knowledge distillation technique does not have control over supervision at the intermediate structured knowledge [45]. So due to the student-teacher representational capacity gap, the students fail to properly mimic the intrinsic distribution [43], [44] of the teacher. For that reason reducing only KL divergence is not enough. Then feature level knowledge distillation-based technique is proposed to supervise the students for mimicking intermediate feature level knowledge such as contrastive representations [21], attention [26], similarity [27], inter-channel diversity-preserved knowledge [28], and so on [21], [38].

Though these works perform better they still suffer from the problem of fidelity [25] and generalization problems. High fidelity means faithfulness to the prediction by the networks, this is related to generalization. If the network works better in training datasets, it is not guaranteed that it will perform better also in testing datasets. If the network is generalized then the network works better in testing datasets.

Fidelity denotes the degree of generalization [25]. The over-fitted models try to memorize the teacher representations in the training dataset, which does not ensure generalization over the test data. Then to improve the generalization and reduce overfitting Xu et al., [38], and Vu et al., [48] proposed techniques to improve the student’s generalization using augmented datasets. Xu et al., [38] and Yang et al., [42] proposed methods to improve teachers by training auxiliary classifiers. We consider transferring highly activated neuron responses via distilling sparse activation maps along with the contrastive knowledge of the augmented and original data.

B. ACTIVATED NEURON’S RESPONSES DISTILLATION
To transfer the neuron’s responses, Romero et al., [30] proposed a knowledge distillation technique that reduces the MSE loss between the neuron’s responses of the teacher and student. Though this technique effectively transfers the neuron’s responses of the high-level layers, it struggles to transfer low-level features. Later, Yim et al., [46] transfer the neuron’s responses by reducing the spatial dimension and the inter-channel correlations are transferred based on the Gram matrix [47]. Zagoruyko et al., [26] reduces channel dimension and focuses on the attention to transfer the neuron’s responses. This technique of reducing dimension happens a loss of information. Heo et al., [33] proposed an activation boundary-based approach without reducing the dimension. Though we also transfer the neuron’s responses, our work is dissimilar to the response-distillation-based techniques in two perspectives: 1) Instead of transferring all the activated neuron’ s responses we transfer the highly activated neuron’s response without reducing dimension to facilitate the transferring of both low and high-level information by shifting the activation boundary, and 2) As the lower activation is dropped we improve generalization using augmented dataset and auxiliary classifiers.

III. PROPOSED NeuRes VIA DISTILLING SAMs
The overall architecture and algorithm of NeuRes are depicted in Fig. 4, and Algorithm. 1, respectively. Proposed NeuRes is composed of four fundamental concerns: A) Feature-level non-linearity, B) Highly activated neuron responses transfer via distilling Sparse Activation Maps (SAMs), C) Translation invariant feature learning via auxiliary classifiers by feeding the augmented data, and D) Deploying NeuRes non-linearity constraints during training the student.

A. FEATURE-LEVEL NON-LINEARITY
1) KEY NOTATIONS
\( \mathcal{A}, \mathcal{S}, \mathcal{F}, \mathcal{N} \), and \( \mathcal{X} \) denote intermediate activation maps after the ReLU activation function, Sparse Activation Maps (SAMs), intermediate feature maps before the ReLU activation function, backbone networks, and datasets, respectively. Moreover, KD loss denotes traditional logits distillation loss proposed by [23].
enhancing feature-level non-linearity. Let’s consider, \( \phi_2 \) FILTER FUNCTION (function.

The filter function \( S \) is done by shifting the separating hyperplane of the activation nodes of lower magnitudes. The candidate selection process is discussed in Section III and Algorithm 1.

This implies that \( A \) includes highly activated neurons’ responses. Let’s assume \( \phi \) through a filter function \( S \) \( i \) \( \mathcal{N}^{\prime}(\cdot) \) and \( \mathcal{N} \) as the student classifier and the teacher classifier, respectively. The classifiers bordered by blue and orange colors indicate the teacher and student classifiers, respectively. The classifiers bordered by blue and orange colors receive the features of the augmented data and main classifiers receive the features of the original data as the input. The purple and orange colors at the top denote the main and auxiliary classifiers, respectively.

The overview of the proposed NeuRes technique. The input data is augmented using four transformations in terms of color, rotation, cropping, and dropping similar to [38]. During training the student, both the teacher and student receive both augmented \( \mathcal{X}^\prime \) and original \( \mathcal{X} \) data as the input. Auxiliary classifiers receive the features of the augmented data and main classifiers receive the features of the original data as the input. The purple and orange colors at the top denote the main and auxiliary classifiers, respectively.

The normalized activation map \( A' \) is then fed into the activation function \( \phi \) to obtain the Sparse Activated Maps (SAMs), \( S_i \). So the filter function can be defined as:

\[
S_i = \phi(A_i)
\]  

The filter function \( S_i \) includes highly activated neurons’ responses by shifting the separating hyperplane of the activation function \( \phi \). In Fig. 1, we show the output and overview of the filter function where the filter function filters out the responses of the low-activated neurons via shifting and scaling the separating hyperplane of the ReLU activation function.

2) FILTER FUNCTION \( \phi \)

Firstly, we normalize the activation map \( A \) into the range of \([-\xi, \xi]\) to obtain the activation map \( A' \) with shifted separating hyperplane. This step shifts the separating hyperplane to re-define the activeness of the neurons where the activated neurons with lower magnitude fall into the negative regions i.e., treated as inactive neurons.

\[
A'_{i,j,k} = \frac{A_{i,j,k} - \text{max}(A)}{\xi \times (\text{max}(A) - \text{min}(A))} \quad |j=0, j=H, k=W
\]  

The normalized activation map \( A' \) is then fed into the activation function \( \phi \) to obtain the Sparse Activated Maps (SAMs), \( S \). So the filter function can be defined as:

\[
S_i = \phi(A_i) = \phi\left(\frac{A_{i,j,k}}{\xi \times ||A||}\right) \quad |j=0, j=H, k=W
\]  

where \( \xi \) denotes the temperature to scale the magnitude of the separating hyperplane. We evaluate the performance of NeuRes for different values of \( \xi \) and experience the best performance at \( \xi = 20 \).

B. HIGHLY ACTIVATED NEURON’S RESPONSES TRANSFER VIA DISTILLING SAMs

The resulting Sparse Activation Maps (SAMs), \( S \) contains highly activated neuron responses. This sparse knowledge is then distilled to transfer the knowledge of the excited neuron’s responses. Let’s assume \( \mathcal{N}^S \) and \( \mathcal{N}^T \) as the student and teacher networks, and the corresponding resulting SAMs are \( S^T \) and \( S^S \), respectively. The Sparse Activation Map \( S^T \) is distilled and the neuron responses are transferred via minimizing the \( L2 \) distance between \( S^T \) and \( S^S \). During training...
the student the L2 loss is used as the regularization term along with the cross-entropy loss. The deployed contraints to perform feature-level non-linearity during transferring the knowledge for the $i$ - $th$ layer is:

$$L^i_{\text{res}}(S^T_i, S^S_i) = ||(S^T_i - S^S_i)||^2_2$$  \hspace{1cm} (4)$$

The total response loss is then:

$$L_{\text{res}}(S^S, S^T) = \sum_{i=1}^{L} ||(S^S_i - S^T_i)||^2_2$$  \hspace{1cm} (5)$$

where $L$ indicates the number of SAMs.

C. TRANSLATION INVARIANT FEATURE LEARNING

Till now we have discussed our proposed NeuRes on the single task (i.e., single classifier) using the original datasets. In this section, we explain the process of achieving richer predictive logits-level knowledge using the augmented data along with the original dataset as the input. Following SSKD [38], we perform four transformations of the original data, 1) cropping, 2) rotation, 3) dropping, and 4) color transformation.

Let’s consider $X$ and $X'$ as the original and transformed datasets, respectively. Firstly, the backbone $N^T_b(\cdot)$ and the classifier $N^T_c(\cdot)$ of the teacher network is trained by imposing only the cross-entropy loss using the original data $X$ such as $N^T_b(N^T_c(X))$. After training, we then freeze the weights of the teacher backbone $N^T_b(\cdot)$ and add an auxiliary classifier $N^T_{ax}(\cdot)$ at the top of the teacher network to learn from the augmented dataset $X'$ such that, $N^T_{ax}(N^T_c(X'))$. This training technique improves the predictive logit-level knowledge of the teacher network. The auxiliary classifiers are trained by deploying the cosine similarity-based contrastive prediction loss between the main $N^T_c(\cdot)$ and auxiliary $N^T_{ax}(\cdot)$ classifiers.

$$C_{i,j} = \cosine(N^T_{ax}(X_i'), N^T_c(X_j))$$  \hspace{1cm} (6)$$

where the cosine function is:

$$C_{i,j} = \frac{\text{dot}(N^T_{ax}(X'_i), N^T_c(X_j))}{\|N^T_{ax}(X'_i)\|_2 \|N^T_c(X_j)\|_2}$$  \hspace{1cm} (7)$$

So the self-supervised loss is defined by:

$$L_{\text{ax-st}} = -\sum_{i} \log \left( \frac{\exp(C_{i,i}/\tau)}{\sum_{j} \exp(C_{i,j}/\tau)} \right)$$  \hspace{1cm} (8)$$

D. DEPLOYING NeuRes CONSTRAINTS AS THE REGULARIZATION TERM

The student network $N^S$ also includes the auxiliary classifier $N^S_{ax}(\cdot)$ along with the basic classifier $N^S_b(\cdot)$ similar to the teacher network. During transferring knowledge the student is forced to mimic the knowledge from both the auxiliary and basic classifiers of the teacher by the corresponding classifiers i.e., $(N^T_c(X) \rightarrow N^S_c(X))$ and $(N^T_b(X) \rightarrow N^S_b(X'))$ by reducing the KL-divergence. The auxiliary classifiers of the student network also learn the self-supervised logits knowledge of the main classifiers of the student i.e., $(N^S_c(X) \rightarrow N^S_{ax}(X'))$. The auxiliary logits distribution distillation loss is defined by:

$$L_{\text{ax-st}} = \tau^2 \sum_{X'} \left( \frac{1}{M-1} \sum_{i=0}^{M-1} p^T_i(X'; \tau) \log(p^S_i(X'; \tau)) \right)$$  \hspace{1cm} (9)$$

The self-supervised loss $L_{\text{ax-st}}$ is estimated by the KD loss between the probability of the instance similarity of the probability matrix achieved by $C_{i,j}$ (Eq. 6) of the two input data examples $X_i$ and $X'_j$. The self-supervision loss and the distillation loss as:

$$L_{\text{ax-st}} = \alpha_1 L_{\text{ax-st}} + \alpha_2 L_{\text{ax-st}}$$  \hspace{1cm} (10)$$

The KD [23] loss to reduce the KL-divergence on the original dataset, $L^S(N^S_b(X))$, and $L^T(N^T_b(X))$, respectively, is as follows:

$$L_{\text{kd}} = \tau^2 \sum_{X'} \left( \frac{1}{M-1} \sum_{i=0}^{M-1} p^T_i(X; \tau) \log(p^S_i(X'; \tau)) \right)$$  \hspace{1cm} (11)$$

where $M$ denotes the number of classes. The overall distillation loss is given by:

$$L_{\text{NeuRes}} = \beta_1 L_{\text{kd}} + \beta_2 L_{\text{ce}} + L_{\text{ax-st}} + L_{\text{res}}$$  \hspace{1cm} (12)$$

Here, $\beta_1$, and $\beta_2$ are the balancing hyperparameters adopted from [38] and $L_{\text{res}}$ indicates the cross-entropy loss.

IV. EXPERIMENTS

A. DATASETS

In this paper, the experimental results are evaluated on three standard benchmark datasets, CIFAR100, STL10, and TinyImageNet. In the CIFAR100 dataset, there is a total of 60,000 labelled images of dimension $(32 \times 32)$ where 50,000 are training and 10,000 are validation images distributed to the total of 100 different classes. The STL10 dataset consists of 5000 training and 800 test images each of resolution $(96 \times 96)$ that are distributed into 10 classes and the dataset is rendered from the ImageNet [50] dataset. TinyImageNet dataset with 200 classes and the small version of the ImageNet includes 100,000 training, 10,000 testing, and 10,000 validation images. The dimension of the images in the TinyImageNet dataset is $(64 \times 64)$. we perform the experimental setup and parameter tuning similar to the setup in [21] and [38]. We run 240 epochs for every experiment except training the auxiliary classifiers of the teachers from which we run for 60 epochs. We investigate the performances of our proposed method on six different architecture setups. Every individual reported result is the average of three individual runs. The initial learning rate for every architecture setup is set to 0.05. The evaluation is performed on an NVIDIA GeForce GTX 1080 GPU. During training, the batch size was 128. Following state-of-the-arts, we select SGD optimizer, and the weight decay rate is $5 \times 10^{-4}$ with a momentum of 0.9. We evaluate the performances for different
Algorithm 1: NeuRes: Highly Activated Neuron’s Responses Distillation

Input: Training data $X = \{(x_i, y_i)\}$, where, $i = 1, 2, 3, \ldots, N$; length of dataset $P$ number of batches in train set $N^T$, and $N^S$ denote trained teacher, and initialized student, respectively;

[Set]: Set $N^T$ into evaluation mode;

[Data Augmentation]: Perform data transformation to obtain augmented data $X^\tau$;

[Train Teacher’s Auxiliary Classifiers]: Add auxiliary classifiers $N_{aux}(\cdot)$ and $N_{aux}^L(\cdot)$, respectively, and train the auxiliary classifiers of the teacher network $N^T$ using self-supervised loss (Eq. 8);

[Neuron’s Responses Distillation]

[Initialize]: $p = 1$;

[Set]: Set $N^S$ into training mode;

while $p \neq P$ do

[Compute]: (Forward Propagation) Calculate the teacher’s activation map $A^T_p$ of $N^T$ for current batch $p$;

[Compute]: (Forward Propagation) Calculate the student activation map $A^S_p$ of student $N^S$ for current batch $p$;

[Normalize]: Normalize both $A^T_p$ and $A^S_p$ into range $[-\xi, \xi]$;

[Activate Neurons]: Impose activation function on both $A^T_p$ and $A^S_p$ to obtain Sparse Activation Maps (SAMs) i.e., $S^T_p$ and $S^S_p$, respectively;

[Compare]: Calculate the neuron’s responses losses $\mathcal{L}_{res}$ (Eq. 5);

[Compute]: Calculate the overall NeuRes loss $\mathcal{L}_{NeuRes}$ (Eq. 12);

[Transfer]: Impose distillation loss function $\mathcal{L}_{NeuRes}$ of the NeuRes between $N^T$ and $N^S$.

[Updating Weights]: (Backward propagation) Perform gradient optimization and update the weights of $N^T$ using the $\mathcal{L}_{NeuRes}$ loss;

[Increase]: Increase $p$ by 1;

end

Output: The resulting trained student $N^S$.

values of $\xi$ and obtain the best performance for $\xi = 20$. We evaluate the performance of the proposed NeuRes on four vision-based tasks, 1) Image classification on the CIFAR100 dataset to verify model compression ability, 2) Transferability on CIFAR100 $\rightarrow$ TinyImageNet and CIFAR100 $\rightarrow$ STL10 dataset pairs, 3) Robustness transfer on adversarial examples on the CIFAR100 dataset, and 4) Few-shot training scenarios.

C. MODEL COMPRESSION

Table 1 shows the significance of the NeuRes on the classification task for model compression. We experience that our method achieves state-of-the-art performances for every similar student-teacher architecture setup. NeuRes consistently outperforms existing state-of-the-art methods. We also notice that the students improve their generalization compared to the baseline approaches. This is due to the improved predictive logits-level knowledge at the auxiliary classifiers of the teacher network. These accuracy improvements indicate that NeuRes successfully improves student performances by transferring neuron responses via distilling sparse activation maps (SAMs).

Fig. 7 shows the comparison of our proposed NeuRes regarding F1-score and confusion matrix compared to logits-level and feature-level state-of-the-art methods, KD [23], and CRD [21], respectively. The figure shows that NeuRes obtains the best F1 score compared to the others. We observe from the confusion matrix that the proposed NeuRes reduces the misclassifications compared to the others.

Metric: Let’s consider, TP, FP, TN, and FN denote the true positive, false positive, true negative, and false negative predictions, respectively.

$$F1 - score = \frac{2 \times Pn \times Re}{Pn + Re}$$

Here, $Pn$ and $Re$ indicate the precision and recall, respectively.

$$Pn = \frac{TP}{TP + FP}$$

And,

$$Re = \frac{TP}{TP + FN}$$

D. ROBUSTNESS TRANSFERABILITY AGAINST ADVERSARIAL ATTACKS

Proposed NeuRes trains the student using sparse activation maps on both original and augmented data that also transfers the robustness to the student. The sparsity and the translation invariant feature learning also improve the robustness of the student. Table 3 shows the significance of NeuRes to improve the robustness of the student against adversarial attacks. Firstly the student is trained using NeuRes approach on the CIFAR100 dataset Then the trained student is evaluated on the perturbed datasets. An adversarial attack is a technique that attempts to fool a trained network with the perturbed image (adding noise to the images). The adversarial attack is performed on the CIFAR100 dataset with the magnitude of $\epsilon = 8/255$ using the Fast Gradient Signed Method (FGSM) [49] method. The WideResNet-40-2 and WideResNet-16-2 are selected as the teacher-student architecture setup. Table 3 indicates that the proposed NeuRes forces the student to perform better than the state-of-the-art in robustness transferability.

E. TRANSFERABILITY OF THE STUDENT ON THE CLEAN DATASETS

As the proposed NeuRes transfers the sparse and highly activated neuron responses, the student achieves better
TABLE 1. Model Compression. Top-1 accuracy comparison among six different architecture setups on the CIFAR100 dataset. The RN and WRN stand for ResNet, and WideResNet, respectively. The VGG denotes very deep networks for image recognition [53]. Bold results represent the best performances. The competing results are quoted from [21]. '*' shows the results of the project provided by [21] that is re-run into our environment.

| Teacher Networks | RN110 | RN110 | RN32x4 | WRN-40-2 | WRN-40-2 | VGG13 |
|------------------|-------|-------|--------|----------|----------|-------|
| Student Networks | *RN32 | *RN20 | RNx8   | WRN-40-1 | WRN-16-2 | VGG8  |
| Teacher (Baseline) | 74.31 | 74.31 | 70.42  | 75.61    | 75.61    | 74.64 |
| Student (Baseline)   | 71.14 | 69.06 | 72.50  | 71.98    | 73.26    | 70.36 |
| *KD [23]  | 73.08 | 70.67 | 73.33  | 74.22    | 75.66    | 72.98 |
| AT [26]   | 72.31 | 70.22 | 73.44  | 72.77    | 74.08    | 71.43 |
| FitNet [30] | 71.06 | 68.99 | 73.50  | 72.24    | 73.58    | 71.02 |
| NST [31]  | 71.96 | 69.53 | 73.30  | 72.24    | 73.68    | 71.53 |
| PKT [32]  | 72.61 | 70.25 | 73.64  | 73.45    | 74.54    | 72.88 |
| SP [27]   | 72.69 | 70.04 | 72.94  | 73.83    | 73.83    | 72.68 |
| AB [33]   | 70.98 | 69.53 | 73.17  | 72.38    | 72.50    | 70.94 |
| CC [34]   | 71.48 | 69.48 | 72.97  | 73.56    | 73.56    | 70.71 |
| RKD [35]  | 71.82 | 69.25 | 71.90  | 72.22    | 73.35    | 71.48 |
| VD [36]   | 72.61 | 70.16 | 73.09  | 73.30    | 74.11    | 71.23 |
| FT [37]   | 72.37 | 70.22 | 72.86  | 71.99    | 73.25    | 70.58 |
| *CRD [21] | 73.48 | 71.46 | 75.51  | 74.14    | 75.48    | 73.94 |
| *SSKD [38] | 73.64 | n/a   | 75.53  | 75.30    | 75.39    | 74.51 |
| NeuRes (Ours) | 73.74 | 71.78 | 76.39  | 75.73    | 76.32    | 75.22 |

TABLE 2. Transferability of students on clean dataset. We use WideResNet-40-2 and WideResNet-16-2 as the teacher and the student architectures, respectively. The competing results are quoted from [21]. Bold values indicate the best performances.

| Transfer | Teacher | Student | KD [23] | AT [26] | FitNet [30] | CRD [21] | SSKD [38] | Ours |
|----------|---------|---------|---------|---------|-------------|----------|----------|------|
| CIFAR-100 → STL-10 | 68.6 | 69.7 | 70.9 | 70.7 | 70.3 | 71.16 | 71.31 | 34.12 |
| CIFAR-100 → TinyImageNet | 31.5 | 33.7 | 33.9 | 33.2 | 33.5 | 33.92 | 25.61 | 25.61 |

TABLE 3. Robustness on adversarial examples of the CIFAR100 dataset. Bold values indicate the best performances. WideResNet-40-2 and WideResNet-16-2 are selected as the teacher and student, respectively. The perturbations are performed with magnitude $\epsilon = 8/255$ using the FGSM method. 'SOTA' stands for state-of-the-art.

| Approaches | Teacher | Student | KD [23] | AT [26] | SP [27] | CRD [21] | SSKD [38] | Ours |
|------------|---------|---------|---------|---------|---------|----------|----------|------|
| Top-1 Accuracy (%) | 26.23 | 22.84 | 24.05 | 22.45 | 23.87 | 22.88 | 24.37 | 25.61 |
| SOTA vs NeuRes ($\Delta$) | n/a | -2.77 | +1.56 | +3.16 | +1.74 | -2.73 | +1.24 | n/a |

TABLE 4. Few-Shot Scenario. Top-1 accuracy (%) comparison of NeuRes on the CIFAR100 dataset. Percentage (%) represents the amount of training data for the particular scenarios. We use ResNet8x4 and ResNet32x4 as the student and teacher, respectively.

| Percentages | KD [23] | CRD [21] | SSKD [38] | Ours |
|-------------|---------|----------|----------|------|
| 25%         | 65.15   | 65.80    | 67.82    | 68.20 |
| 50%         | 68.61   | 69.91    | 70.09    | 70.88 |
| 75%         | 70.34   | 70.98    | 70.47    | 71.03 |

From Table 2 we experience that, NeuRes outperforms existing state-of-the-art works in transferability tasks. The results in Table 2 indicate that sparsity and highly activated neuron responses along with translation invariant feature learning transfer the robustness to the student.

F. ATTENTION COMPARISON: CLASS ACTIVATION MAP
Our proposed method NeuRes is highly related to attention. To evaluate the significance of our proposed method in terms of focusing on the interesting regions i.e., optimum attention, we investigate the class activation map of the trained student. From Fig. 8 we observe that the proposed method successfully pays attention to the expected regions to provide better results compared to the KD [23] knowledge distillation method.

G. SIGNIFICANCE IN FEW-SHOT TRAINING SCENARIO
Table 4 and Fig. 5 show that our proposed NeuRes outperforms the state-of-the-art methods in the few-shot training scenario. The training is performed on three different few-shot scenarios, we feed 25%, 50%, and 75% training data and the original test set of the CIFAR100 dataset to evaluate the significance of the NeuRes in few-shot training. More specifically, if the dataset contains a total of $N$ number of...
images, then 25%, 50%, and 75% of training data indicate that $0.25 \times N$, $0.5 \times N$ and $0.75 \times N$ number of images, respectively, are used during training. We perform the same dataset split mechanism similar to SSKD [38]. The ResNet8x4 and ResNet32x4 are selected as the student and teacher networks, respectively. From Table 4 we see that

**FIGURE 5.** Training and testing curves in the few-shot training scenarios for 25% and 50% of the CIFAR100 dataset. The ResNet8x4 and ResNet32x4 are used as the student and teacher, respectively.

**FIGURE 6.** t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization: We show the effectiveness in estimating decision boundaries on the CIFAR100 test dataset. ResNet32x4 and ResNet8x4 are considered as the teacher, and the student, respectively.

**FIGURE 7.** Confusion matrix and F1-score comparison.
the proposed NeuRes outperforms existing state-of-the-art methods in few-shot training in all the scenarios.

We notice from Table 4 that our method NeuRes can achieve comparable performance using only 25% of the training dataset. The reported results in Table 4 verify the effectiveness of the NeuRes in few-shot training.

H. EFFECTS ON DECISION BOUNDARY
We validate the quality of global features and the ability to match the teacher decision boundary of the proposed method through t-SNE visualization. The t-distributed stochastic neighbor embedding (t-SNE) is a method to visualize high-dimensional data in a two or three-dimensional space by performing dimensional reduction using a statistical approach. Fig. 6 shows the visualization of the teacher, KD [23] and NeuRes. ResNet32x4 and ResNet8x4 are selected as the teacher, and the student, respectively. The visualization is obtained from the pretrained corresponding student network. We evaluate the performance in improving decision boundaries on the CIFAR100 test dataset. From Fig. 6 we notice that our proposed method improves the decision boundary with compare to the original KD [23] method. We see that the decision boundary of ‘Ours’ method is more similar to the teacher compared to the baseline KD [23]. This outcome further proves the capacity of the proposed technique and the NeuRes improves the decision boundary by deploying activation boundary shifting and scaling.

V. CONCLUSION
We propose a novel knowledge distillation technique, called NeuRes to transfer the highly activated neuron’s responses along with translation invariant features via distilling the Sparse Activation Maps (SAMs). The proposed technique, with the help of auxiliary classifiers and augmented data improves the representational capability as well as predictive logits knowledge of the teacher network to perform better knowledge distillation and transfer. NeuRes outperforms existing state-of-the-art results in transferring knowledge to the student. The proposed NeuRes loss \( \mathcal{L}_{NeuRes} \) can be deployed as the assistant performance enhancer to any knowledge distillation techniques for further improving student performances. We validate the effectiveness of the proposed technique through extensive experiments on model compression, transferability, robustness transfer, and few-shot training scenarios. Our proposed method requires an identical spatial dimension of the teacher and student feature maps. In our future work, we will investigate transferring knowledge to students of different architecture.

REFERENCES
[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1–9.
[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.
[4] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao, “NeuralRecon: Real-time coherent 3D reconstruction from monocular video,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 15598–15607.
[5] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, “A survey and performance evaluation of deep learning methods for small object detection,” Expert Syst. Appl., vol. 172, Jun. 2021, Art. no. 114602.
[6] S. S. A. Zaidi, M. S. Ansari, A. Aslam, A. Khan, M. A. Asghar, and B. Lee, “A survey of modern deep learning based object detection models,” Digit. Signal Process., vol. 126, Jun. 2022, Art. no. 103514.
[7] X. Yuan, J. Shi, and L. Gu, “A review of deep learning methods for semantic segmentation of remote sensing imagery,” Expert Syst. Appl., vol. 169, May 2021, Art. no. 114417.
[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861.
MD IMTIAZ HOSSAIN received the B.Sc. and M.Sc. degrees from the Department of Information and Communication Technology (ICT), Islamic University, Bangladesh, in 2016 and 2018, respectively, and the M.S. degree from the Department of Computer Science and Engineering, Kyung Hee University, Global Campus, Yongin-si, South Korea, in 2021, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. He has been a Research Student with the Intelligent Computing and Security Laboratory (ICNS Lab), Kyung Hee University, since 2019. His research interests include transfer learning, knowledge distillation, visual-based human action recognition, 3D pose estimation, 3D motion modeling, and the metaverse.

MD DELOWAR HOSSAIN received the B.Sc. and M.Sc. degrees from the Department of Information and Communication Engineering (ICE), Islamic University, Bangladesh, in 2004 and 2005, respectively, and the Ph.D. degree from the Department of Computer Science and Engineering, Kyung Hee University, Republic of Korea. He was a Visiting Scholar with Infosys, Bengaluru, India. He is currently working as a Professor with the Department of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh, where he was the Chairman, from 2011 to 2013. He is also working as a Postdoctoral Researcher with the Department of Computer Science and Engineering, Kyung Hee University. His current research interests include cloud/edge/fog computing, vehicular edge computing, big data, machine learning, and the Internet of Things. He was a recipient of the Best Paper Award from KSC 2018, KSC 2019, and KCC 2021, South Korea.

EUI-NAM HUH (Member, IEEE) received the B.S. degree from Busan National University, South Korea, the master’s degree in computer science from The University of Texas, Austin, TX, USA, in 1995, and the Ph.D. degree from Ohio University, Athens, OH, USA, in 2002. He is currently a Professor with the Department of Computer Science and Engineering, Kyung Hee University, South Korea. His research interests include cloud computing, the Internet of Things, future internet, distributed real-time systems, mobile computing, big data, and security. He is a Review Board of the National Research Foundation of Korea. He has also served many community services for ICCSA, WPDRTS/IPDPS, APAN Sensor Network Group, ICUIMC, ICONI, APIC-IST, ICUFN, and SoICT as various types of chairs. He is the Vice-Chairman of the Cloud/Bigdata Special Technical Group of TTA and an Editor of ITU-T SG13 Q17.