IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 9, 2021, accepted April 27, 2021, date of publication May 5, 2021, date of current version May 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077596

Compressing Neural Networks With Inter Prediction and Linear Transformation

KANG-HO LEE AND SUNG-HO BAE, (Member, IEEE)
Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, Republic of Korea

Corresponding author: Sung-Ho Bae (shbae@khu.ac.kr)

This work was supported by the Ministry of Trade, Industry, and Energy (MOTIE), South Korea, through the Technology Innovation Program or the Industrial Strategic Technology Development Program (Memristor Fault-Aware Neuromorphic System for 3-D Memristor Array) under Grant 10085646.

ABSTRACT Because of resource-constrained environments, network compression has become an essential part of deep neural networks research. In this paper, we found a mutual relationship between kernel weights termed as *Inter-Layer Kernel Correlation* (ILKC). The kernel weights between two different convolution layers share a substantial similarity in shapes and values. Based on this relationship, we propose a new compression method, *Inter-Layer Kernel Prediction* (ILKP), which represents convolutional kernels with fewer bits through similarity between kernel weights in convolutional neural networks. Furthermore, to effectively adapt the inter prediction scheme from video coding technology, we integrate a linear transformation into the prediction scheme, which significantly enhances compression efficiency. The proposed method achieved 93.77% top-1 accuracy with 4.1× compression ratio compared to the ResNet110 baseline model on CIFAR10. It means that 0.04% top-1 accuracy improvement was achieved by using less memory footprint. Moreover, incorporating quantization, the proposed method achieved a 13× compression ratio with little performance degradation compared to the ResNets baseline model trained on CIFAR10 and CIFAR100.

INDEX TERMS Deep learning, model compression, neural networks, neural network compression, inter prediction, linear transform, quantization, ILKC, ILKP, ILKP-Q.

I. INTRODUCTION

Recently, Deep Neural Networks (DNN), specifically Convolutional Neural Networks (CNN), are showing exceptional performance compared with traditional methods for a wide variety of tasks in many fields such as image classification [1]–[3], object detection [4]–[6], and also speech recognition [7], [8]. However, with this performance improvement, the size of the CNN model has increased enormously, and the recent works are also expanding in size with more model parameters for better performance.

These large CNN models, which could not be driven only in the early 2000s, became possible with the advancement of hardware. However, it is still challenging to deploy them in an environment with limited computing resources such as mobile environment, embedded system environment, navigation system and a kiosk.

In order to solve this problem, methods of reducing the size of a CNN model or designing an efficient CNN structure have emerged as core fields in recent deep neural network research. Representative methods include pruning [9]–[12], quantization [13]–[15], knowledge distillation [16]–[18], weight sharing [19]–[22], and efficient structural design methods, e.g., Depthwise Separable Convolution [23]–[26]. These methods are widely used to compress the size of the CNN model.

Among them, the weight sharing methods are categorized into: i) vector quantization methods where the similar-valued weights are clustered into classes [15], [27]; ii) and methods that design the networks to have the shared weights [19], [28]. It has been shown that both methods drastically reduce the number of CNN parameters as compared to previous papers. Contrary to the existing weight sharing methods, we propose a new weight sharing method based on prediction techniques essentially used in traditional video compression technology, such as H.264 [29] and H.265 [30].

Prediction methods are based on the fact that there are many parts with high correlations within each intra-frame as well as between inter-frames in sequential frames. For prediction, each frame is divided into macro-blocks (MBs),

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

and each MB in the current frame is predicted by searching the best MB in the search range with respect to minimizing rate-distortion costs. As the residuals between the current and predicted (best) MBs tend to have low entropy following a narrow-shaped Laplacian distribution, the residuals are compressed and stored in the encoder side [29], [30].

In this paper, we show that the kernel[1] weights between the two different convolution layers tend to share high similarity in shapes and values. We call this reciprocal relationship *Inter-Layer Kernel Correlation* (ILKC). This paper explores an efficient neural network method that fully takes advantage of the ILKC hypothesis. Based on ILKC, this paper proposes a simple and effective weight compression method *Inter-Layer Kernel Prediction* (ILKP) that effectively shares the weights by prediction.

A. CONTRIBUTIONS

Main contributions of this paper are listed below:

- From comprehensive experiments, we find out that the kernel weights between the two different convolutional layers in a CNN tend to share strong similarities, which lead us to hypothesizing ILKC.
- Based on ILKC, we propose a simple and useful model weight compression method, ILKP that minimizes the weight sizes by prediction.
- The proposed ILKP achieves about 4.1× compression ratio on average at the same accuracy level compared to the ResNets [3] (ResNet 20/32/44/56/110) baseline model on CIFAR10 and CIFAR100.
- Furthermore, by combining an efficient quantization method, the proposed method, called Quantized ILKP (ILKP-Q), achieves about 13× compression ratio on average with little accuracy degradation compared to the baseline model (without prediction) in various ResNet models [3] (ResNet 20/32/44/56/110) trained on CIFAR10 and CIFAR100.

II. RELATED WORK

A. NETWORK PRUNING

Network pruning methods prune the unimportant weight parameters, enabling to reduce the redundancy of weight parameters inherent in neural networks. References [9] and [10] reduced the number of weight connections implicitly through setting a proper objective function for training. Reference [11] successfully removed the unimportant weight connections through certain thresholds for the weight values, showing no harm of accuracy in the state-of-the-art convolutional neural network models. Recently, structured (filter/channel/layer-wise) pruning methods have been proposed in [31] and [12], where a set of weights is pruned based on certain criteria (e.g., the sum of absolute values in the set of weights), demonstrating significantly reduced number of weight parameters and computational costs. Furthermore, [32] uses AutoML for channel pruning and their proposed method yields 13.2% higher accuracy than filter pruning method [12]. Our paper is linked with removing unimportant weights of the pruning method to prevent the use of inessential weights in the kernel by utilizing the kernel on the previous layer using the prediction method.

B. QUANTIZATION

Quantization reduces the representation bits of original weights in neural networks. Reference [13] proposed a weight quantization using weight discretization in neural networks. Reference [11] incorporated a vector quantization into pruning, proving that quantization and pruning can jointly work for weight compression without accuracy degradation. Deep Compression, a pruning-quantization framework, became a milestone in model compression research of deep neural networks. Reference [33] proposed a fixed-point quantization using a linear scale factor for weight values where bit-widths for quantization are adaptively found for each layer, enabling a 20% reduction of the weight size in memory without any loss in accuracy compared to the baseline fixed-point quantization method. Furthermore, [34], [35] and [36] use clipping weights before applying linear quantization, thus improving accuracy compared to linear quantization without clipping. In this paper, we apply a quantization technique to the by-product of predictions (e.g., indices of predicted weight kernels), thus dramatically compressing the parameter sizes of CNNs.

C. WEIGHT SHARING

In Deep Compression [15], the CNN model was compressed by collecting similar weights in each layer and quantizing them where similar-valued weight values are clustered into few classes, which can be viewed as a weight sharing in values. In BSConv [40], it showed that similar kernels exist in each layer, and scalar multiplication was used on one kernel shared in each filter map in a vanilla convolutional layer. They proved that through equation rearrangement, the weight sharing property can make a convolution layer decomposed into a point-wise convolution layer and a depth-wise convolution layer.

In the above studies, weight sharing was performed based on the existence of similar weights within layers. However, this paper found that similar kernels exist between two different layers. Therefore, weight sharing and model compression method based on the inter-frame prediction method, that is, ILKP is proposed accordingly.

D. PREDICTION IN CONVENTIONAL VIDEO CODING

Prediction techniques are considered one of the most crucial parts of video compression, aiming at minimizing the magnitudes of signals to be encoded by subtracting the input signals to the most similar encoded signals in a set of prediction candidates [29], [30], [41]. The prediction methods can produce the residuals of signals with low magnitudes and a

[1] In this paper, the kernel denote as a two-dimensional convolution kernel obtained by dividing a three-dimensional convolution filter map channel-wise, e.g., in 3 × 3×32 convolution filter map has 32 3 × 3 kernels.

large number of non/near zero signals. Therefore, they have effectively been incorporated into transforms and quantization for concentrating powers in low frequency regions and reducing the entropy, respectively. There are two prediction techniques: inter- and intra- predictions. The inter-prediction searches the best prediction signals from the encoded neighbor frames out of the current frame. At the same time, the intra-prediction generates a set of prediction signals from the input signals and determine the best prediction [29], [30]. This is because intra frames that use only intra-prediction for compression are used as reference frames for subsequent frames to be predicted.

Note that a few studies explored to apply the transform techniques of video and/or image coding to the weight compression problem in neural networks. References [42] and [43] applied DCT (Discrete Cosine Transform) used in the JPEG (Joint Picture Encoding Group) algorithm to the model compression problem of deep neural networks such that the energy of weight values became concentrated in low frequency regions, thus producing more non/near-zero DCT coefficients for the weights. Compared to the papers mentioned above, our work does not adopt transform techniques to reduce model sizes since transformations introduce high computational cost during inference, decreasing the effectiveness of the weight compression in practical applications.

In this paper, we found out that the inter-prediction technique can play a crucial role for weight compression. As a result, the proposed inter-prediction method for memory yields impressive compression performance enhancement at the similar accuracy level compared to the baseline models.

III. INTER-LAYER KERNEL CORRELATION (ILKC)

From the perspective of prediction technique in video compression, two statistical characteristics of signals are essential to have high prediction performance, i.e., the similarities in magnitude range and direction between two signals.

To show similarity in magnitude range of weights among layers, we plot the weight distribution in each layer obtained from the pre-trained ResNet20 on CIFAR10 in Figure 1. As shown in Figure 1, the weight values tend to be in certain ranges regardless of their layer positions, supporting our ILKC hypothesis.

To observe the similarity in direction of weight kernels between layers, we measure the maximum absolute value of Pearson Correlation Coefficient (MA-PCC) between a kernel in the i-th current layer (C_i) and a set of all kernels in the j-th reference layer (R_j). The MA-PCC that is defined for the l-th kernel in C_i and is calculated as

$$\text{MA-PCC}(C_{i,l}, R_j) = \max_{k \in \{1,\cdots,K\}} \left| r(C_{i,l}, R_{j,k}) \right|, \qquad (1)$$

where $C_{i,l}$ is the l-th kernel in C_i, $R_{j,k}$ is the k-th kernel in R_j, and K is the number of kernels in R_j. In Eq. (1), $r(C_{i,l}, R_{j,k})$ is the value of the sample Pearson Correlation Coefficient

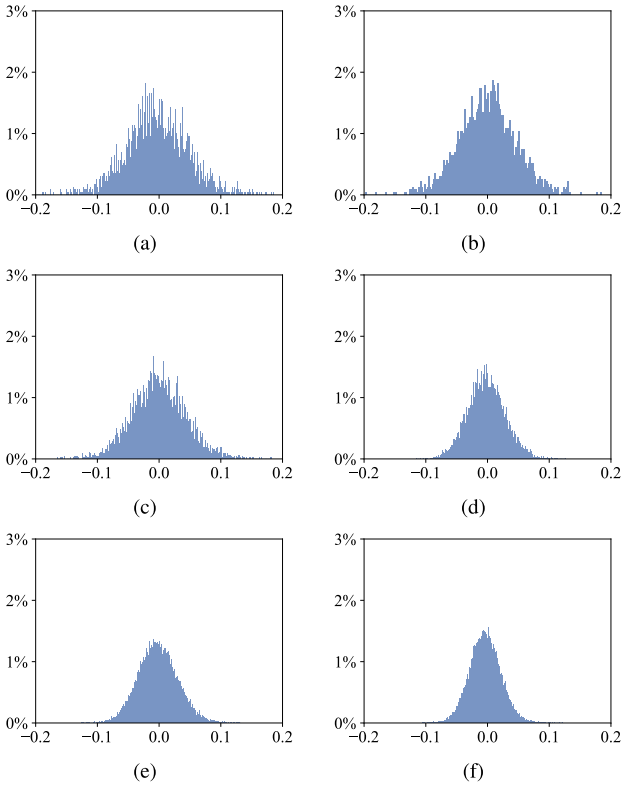

FIGURE 1. Weight distribution of each convolutional layer in ResNet20 trained on CIFAR10. (a) 2nd layer; (b) 4th layer; (c) 7th layer; (d) 12th layer; (e) 14th layer; (f) 17th layer; Layer ordinal numbers start at 0.

(sample PCC; PCC) [44] which is calculated as

$$r(X, Y) = \frac{\sum \left(X_m - \bar{X}\right)\left(Y_m - \bar{Y}\right)}{\sqrt{\sum \left(X_m - \bar{X}\right)^2 \sum \left(Y_m - \bar{Y}\right)^2}}, \qquad (2)$$

where X_m (or Y_m) is the m-th element when each kernel is vectorized, and $\bar{X}$ (or $\bar{Y}$) is the mean of X (or Y).

Figure 2 and 3 show the MA-PCC for combination of different layers in pre-trained ResNet20 on CIFAR10 and CIFAR100. As shown in Figure 2 and 3, overall, MA-PCC values are high enough, indicating that the kernels between two different convolutional layers are similar in direction. Therefore, based on ILKC, we propose an inter-layer prediction method in the next section.

IV. INTER-LAYER KERNEL PREDICTION

A. INTER-LAYER KERNEL PREDICTION (ILKP)

We propose ILKP motivated from the inter-frame prediction method. As shown in Figure 4, the ILKP is a method of approximating the current kernel with a linear transformed reference kernel with a scale factor (α) and offset factor (β). It is noted that the search range for reference layers directly affects training time. From our comprehensive experiments, we found that the first layer of the neural network (i.e., R_0) with linear transformation (LT) tends to well approximate the dominant portion of rest kernels in the whole network. Therefore, using R_0 as a reference layer significantly reduces

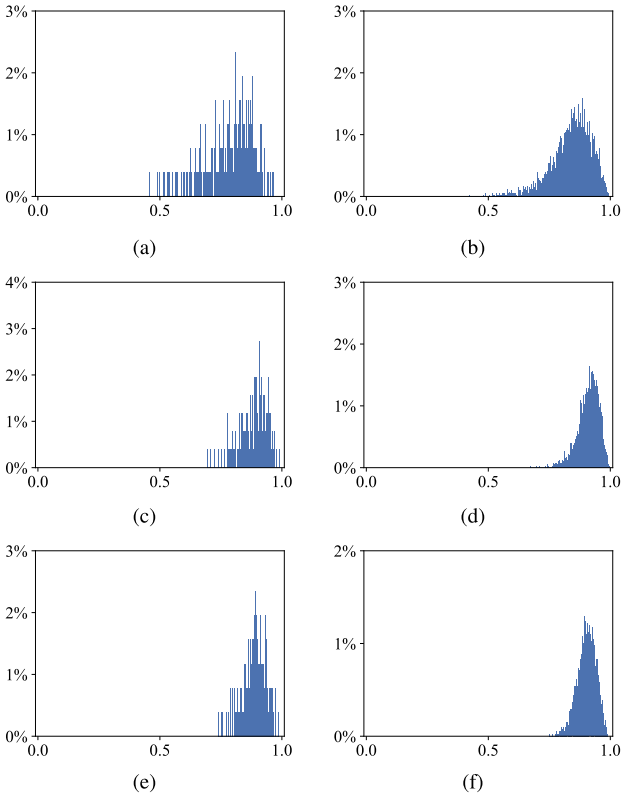

FIGURE 2. Histogram of MA-PCC values between two different convolutional layers in ResNet20 trained on CIFAR10. Between (a) R_0 and C_6; (b) R_0 and C_{18}; (c) R_1 and C_6; (d) R_1 and C_{18}; (e) R_5 and C_6; (f) R_5 and C_{18}.

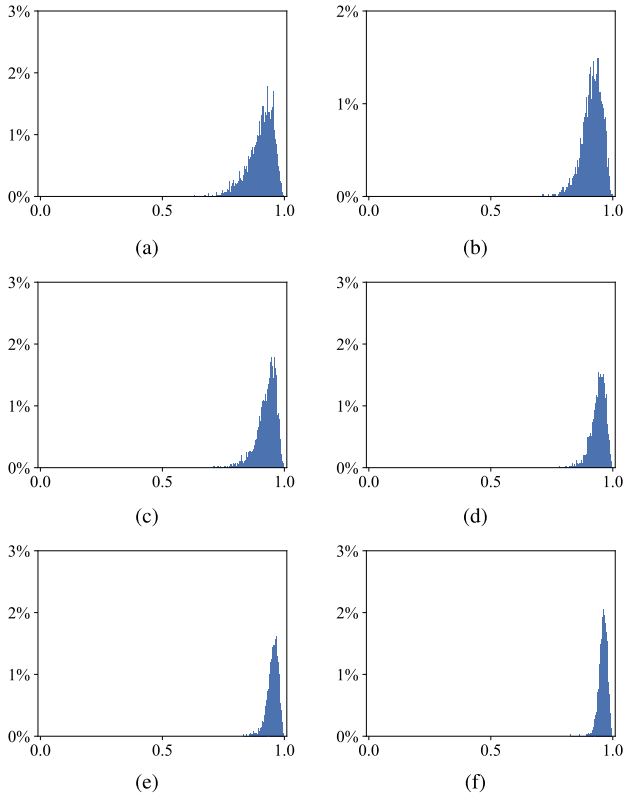

FIGURE 3. Histogram of MA-PCC values between two different convolutional layers in ResNet20 trained on CIFAR100. Between (a) R_7 and C_{16}; (b) R_7 and C_{18}; (c) R_{12} and C_{16}; (d) R_{12} and C_{18}; (e) R_{15} and C_{16}; (f) R_{15} and C_{18}.

training time. During training, the proposed method updates the weights in the reference layer as well as dynamically searches the most similar kernel (denoted as the best kernel) to the current kernel from the reference layer in each epoch with respect to maximizing absolute PCC as

$$\hat{k} = \underset{k\in\{1,\cdots,K\}}{\arg\max} \left| r(C_{i,l}, R_{0,k}) \right|. \tag{3}$$

After finding the best reference kernel X, α and β for the current kernel Y are calculated with the least-squares estimate of the slope (α) and the intercept (β) [45] as

$$\begin{cases} \alpha = \dfrac{\sum \left(X_m - \bar{X}\right)\left(Y_m - \bar{Y}\right)}{\sum \left(X_m - \bar{X}\right)^2} \\ \beta = \bar{Y} - \alpha\bar{X}. \end{cases} \tag{4}$$

Finally, we replace the current kernel with the three elements, i.e., α, β and the index of the best kernel ($\hat{k}$) which are stored in memory. Note that $\hat{k}$ has a bit-depth of $\lceil \log_2 n \rceil$ where n is the number of kernels in the reference layer. The ILKP process can be seen in Algorithm 1.

During the inference time, only the kernels in the reference layers and a set of α, β and $\hat{k}$ are accessed from the memory by which all weights in the network can be restored.

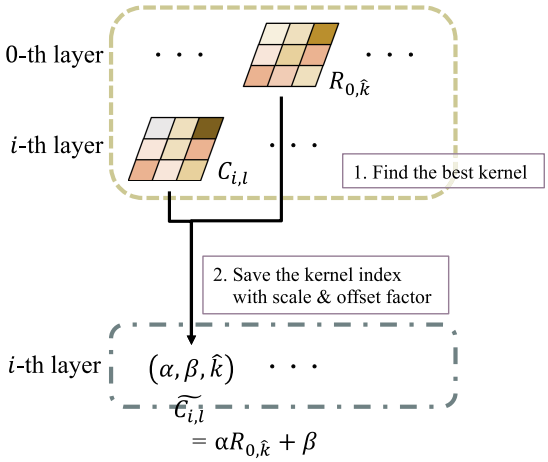

FIGURE 4. An illustration of the proposed method. The upper dashed box represents the conventional CNN where $R_{0,\hat{k}}$ and $C_{i,l}$ are the $\hat{k}$-th kernel in the 0-th layer and the l-th kernel in the i-th layer, respectively. The lower one-dot chain box represents ILKP parameters of $C_{i,l}$ by prediction with $R_{0,\hat{k}}$.

B. ILKP WITH α, β QUANTIZATION (ILKP-Q)

To maximize the effectiveness of the proposed ILKP, we propose ILKP-Q which quantizes α and β. For training the quantized α and β, we apply the quantization-aware training

Algorithm 1 ILKP

Input: Pretrained model
Output: ILKP model
// N is the number of layers
// $M(i)$ is the number of kernels in i-th layer
for $i = 1$ to $N - 1$ **do**
  for $l = 0$ to $M(i) - 1$ **do**
    find $\hat{k}$ using Eq. (3)
    calculate α, β between $C_{i,l}$ and $R_{0,\hat{k}}$ using Eq. (4)
    store $(\alpha, \beta, \hat{k})$
  end for
end for
return R_0 and $(\alpha, \beta, \hat{k})$ list

method [46]. That is, the full-precision α and β are kept in memory where the quantized α and β are used for feed-forward process, and the gradients are updated with full-precision α and β during back-propagation. The quantization method used in this experiment is a linear uniform quantization, and α and β are quantized to 8 bits each epoch. Since the α and β of each kernel are reduced from 32 to 8 bits, much higher compression ratio can be obtained.

V. EXPERIMENTAL RESULTS

A. EXPERIMENTAL DETAILS

In this section, we describe and prove the superiority of the proposed method by applying it on image classification tasks, specifically for CIFAR10 and CIFAR100 datasets. For securing the generality of our proposed ILKP, we applied our method in pre-trained ResNet20/32/44/56/110 models. The framework used in the experiment is PyTorch. One NVidia 2080-Ti GPU with the Intel i9-7900X CPU is used to perform the experiments. For the hyper-parameter setting in the training process, we set the initial learning rate as 0.01, which is multiplied by 0.98 after every epochs. We used Stochastic Gradient Descent (SGD) optimizer with Nesterov momentum [47] factor 0.9. All the neural networks are trained for 200 epochs with a batch size of 256. In all the experiments, the test accuracy and compression ratio are marked from the average of 5 runs. Compression ratio (CR) is computed by the ratio of the total number of bits of the convolution weights for the baseline over that of the proposed methods.

B. INTER-LAYER KERNEL PREDICTION

As shown in Table 1, the proposed ILKP decreases the sizes of all the test models more than 4× with negligible performance drop compared to the baseline models. In particular, the largest test model, i.e., ResNet110, it obtained a compression ratio of 4.11× with a performance increase of 0.04% on CIFAR10.

C. ILKP WITH α, β QUANTIZATION (ILKP-Q)

We show the experimental results of ILKP-Q on α and β 8-bit quantization. As shown in Table 2 and Table 3, ILKP-Q achieved a remarkable compression ratio of about 13× compared to Baseline and about 3× compared to the original ILKP based models with negligible performance degradation.

For better visibility purpose, we plot the performance curves in trade-off between top-1 accuracy and the total weight sizes in convolution layers of tested models on CIFAR10 and CIFAR100 in Figure 5, where 'Baseline-Q' is the quantized model with a linear quantization [46] in the baseline model for {8, 7, 6, 5, 4} weight bits. As shown in Figure 5, in most cases, ILKP has a slightly lower or almost similar top-1 accuracy at the similar levels of compression ratios to Baseline-Q. In addition, it is shown that, as the model size increases, ILKP gradually improves the performance in trade-off compared to Baseline-Q. This is because the proposed ILKP keeps only one set of weight kernels in the reference layer while the rest are replaced with ILKP parameters $(\alpha, \beta, \hat{k})$. The ILKP-Q shows superior performance in a trade-off between top-1 accuracy and compression ratio to both Baseline-Q and ILKP, especially showing better performance on CIFAR100.

TABLE 1. Test top-1 accuracy and compression ratio of the proposed method compared to the baseline.

Dataset	Model	Top-1 accuracy (%)		CR (×)
		Baseline	Ours	
CIFAR10	ResNet20	92.27	91.25±0.16	4.09
	ResNet32	92.66	92.19±0.17	4.10
	ResNet44	93.24	93.13±0.19	4.11
	ResNet56	93.52	93.24±0.19	4.11
	ResNet110	93.73	93.77±0.18	4.11
CIFAR100	ResNet20	66.54	65.70±0.30	4.09
	ResNet32	68.96	67.70±0.11	4.10
	ResNet44	69.57	68.15±0.10	4.11
	ResNet56	70.62	69.41±0.29	4.11
	ResNet110	72.85	71.47±0.18	4.11

TABLE 2. Test top-1 accuracy and compression ratio of ILKP and ILKP-Q compared to the baseline trained on CIFAR10. 'Base' means baseline.

Model	Top-1 accuracy (%)			CR (×)	
	Base	ILKP	ILKP-Q	ILKP	ILKP-Q
ResNet20	92.27	91.25±0.16	89.00±0.42	4.09	12.84
ResNet32	92.66	92.19±0.17	92.07±0.10	4.10	12.94
ResNet44	93.24	93.13±0.19	92.93±0.18	4.11	12.99
ResNet56	93.52	93.24±0.19	93.09±0.09	4.11	13.01
ResNet110	93.73	93.77±0.18	93.13±0.23	4.11	13.05

TABLE 3. Test top-1 accuracy and compression ratio of ILKP and ILKP-Q compared to the baseline trained on CIFAR100. 'Base' means baseline.

Model	Top-1 accuracy (%)			CR (×)	
	Base	ILKP	ILKP-Q	ILKP	ILKP-Q
ResNet20	66.54	65.70±0.30	65.66±0.27	4.09	12.84
ResNet32	68.96	67.70±0.11	67.76±0.27	4.10	12.94
ResNet44	69.57	68.15±0.10	67.68±0.39	4.11	12.99
ResNet56	70.62	69.41±0.29	68.54±0.27	4.11	13.01
ResNet110	72.85	71.47±0.18	69.47±0.70	4.11	13.05

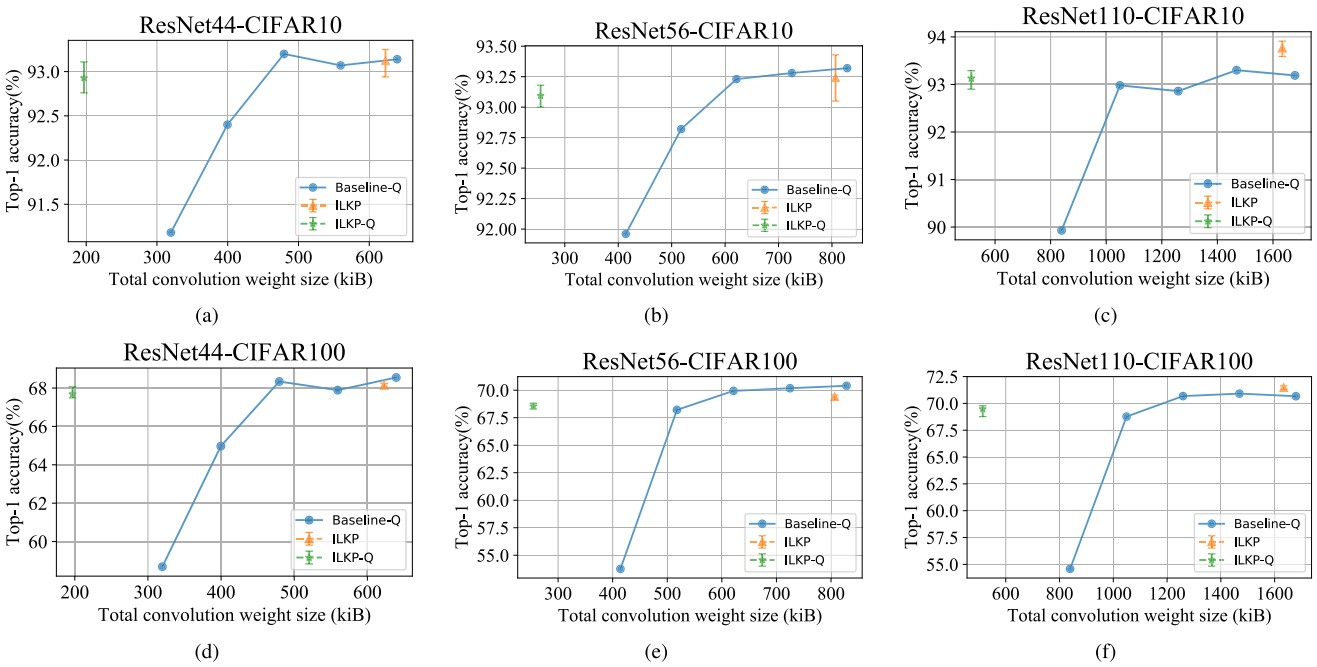

FIGURE 5. Comparison between proposed methods with linear quantization method on baseline ResNet44/56/110 trained on CIFAR10/100.

TABLE 4. Test top-1 accuracy of ablation studies on CIFAR10.

Model	Top-1 accuracy (%)		
	ILKP w/o α, β	ILKP w/o β	ILKP
ResNet20	62.41±2.61	88.36±0.17	**91.25**±0.16
ResNet32	63.65±4.81	89.84±0.26	**92.19**±0.17
ResNet44	65.70±1.02	90.82±0.18	**93.13**±0.19
ResNet56	66.16±2.33	91.55±0.45	**93.24**±0.19
ResNet110	72.98±0.56	92.65±0.16	**93.77**±0.18

TABLE 5. Test top-1 accuracy of ablation studies on CIFAR100.

Model	Top-1 accuracy (%)		
	ILKP w/o α, β	ILKP w/o β	ILKP
ResNet20	23.79±7.69	54.36±3.81	**65.70**±0.30
ResNet32	27.17±0.84	64.16±0.24	**67.70**±0.11
ResNet44	23.62±7.43	65.89±0.51	**68.15**±0.10
ResNet56	28.63±3.30	67.02±0.15	**69.41**±0.29
ResNet110	31.06±3.73	68.86±0.41	**71.47**±0.18

D. ABLATION STUDY OF ILKP

For ablation studies, we investigated why LT parameters are needed. We apply the proposed ILKP with and without α and β. At this time, finding a similar kernel is the same as the method in section IV-A. It is noted that, under no bias condition, the least squared method for estimating α is calculated as

$$\alpha = \frac{\sum X_m Y_m}{\sum X_m^2}, \tag{5}$$

where the notations are equal to Eq. (4).

As shown in Table 4 and Table 5, LT with α and β play crucial roles in ILKP.

E. ANALYSIS

Figure 6 shows the confusion matrices of baseline and ILKP on CIFAR10. indicating that ILKP often shows higher accuracy compared to the baseline over classes.

To further investigate the effectiveness of the proposed method, we evaluate the similarity in average PCC between reference and current kernels. Figure 7 compares Similarity in average PCCs in ResNet20 trained on CIFAR10 where three modes are experimented: 1) Random prediction: for all kernels of each layer in the pretrained baseline model, reference kernel is randomly selected and the absolute PCC values between the reference and current kernels are calculated; 2) ILKP: the reference kernel is obtained with Eq. (3) in ILKP, and absolute PCC is calculated between the reference and current kernels; 3) ILKP + LT: the reference kernel is obtained with Eq. (3) in ILKP and the LT is applied to the reference kernel to approximate the current kernel. The absolute PCC is calculated between the reference with LT and current kernels.

As shown in Figure 7, the average PCC is lower than 0.5 when the reference kernel is randomly selected. When ILKP with Eq. (3) is applied, the average PCC is increased by 0.3 points. Moreover, it is shown that LT allows to approximating the current kernel very well from the reference kernel, achieving almost 1 in average PCC. This clearly supports the effectiveness of the proposed ILKP with LT.

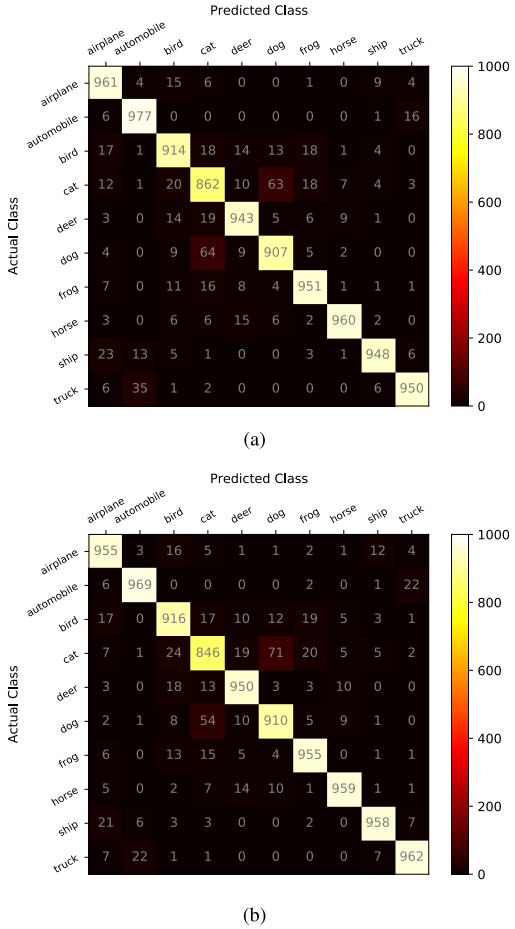

FIGURE 6. Confusion matrices of ResNet110 trained on CIFAR10; (a) Baseline (b) ILKP.

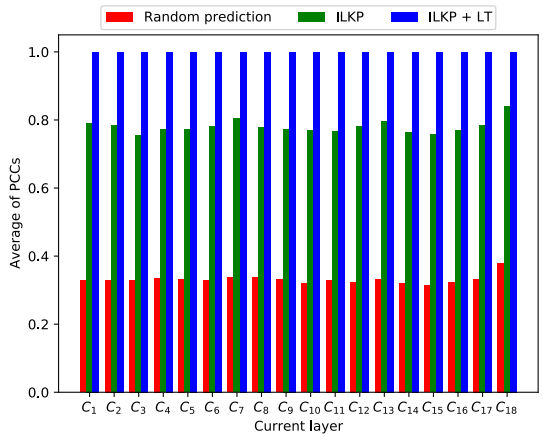

FIGURE 7. Similarity comparison in PCCs in ResNet20 trained on CIFAR10. 1) Random prediction: for all kernels of each layer in the pretrained baseline model, reference kernel is randomly selected and the absolute PCC values between the reference and current kernels are calculated; 2) ILKP: the reference kernel is obtained with Eq. (3) in ILKP, and absolute PCC is calculated between the reference and current kernels; 3) ILKP + LT: the reference kernel is obtained with Eq. (3) in ILKP and the LT is applied to the reference kernel to approximate the current kernel; The absolute PCC is calculated between the reference with LT and current kernels.

F. LIMITATION

Although the proposed method shows its effectiveness in many ResNet models with identical spatial resolutions for all kernels, it cannot be applied to more complex models consisting of various kernel shapes and sizes. In future work, we will resolve this mismatch problem between reference and current kernels.

VI. CONCLUSION

We propose a new inter-layer kernel prediction method for efficient deep neural networks. Motivated by our observation that the kernels between the layers tend to have high similarity, we successfully build a new weight compression framework using the inter-layer kernel prediction scheme. To the best of our knowledge, this work is the first to exploit the mutual relationship of the kernel similarities between the convolutional layers in the context of the inter prediction method in modern video coding technology. Furthermore, to effectively apply the conventional inter prediction method into the weight compression scheme in neural networks, we devise a LT and quantization scheme which significantly enhance the compression efficiency. Our comprehensive experiments show that ILKP-Q achieves outstanding compression efficiency compared to the baseline models. As future work, the proposed method can further be extended for wider applicability by generalizing the prediction scheme for various kernels with different sizes and shapes.

ACKNOWLEDGMENT

The authors are very grateful to Muhammad Salman Ali for checking the grammar of the final article.

REFERENCES

[1] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[3] 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.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[6] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Oct. 2012.

[8] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 conversational speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5934–5938.

[9] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 598–605.

[10] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Proc. Adv. Neural Inf. Process. Syst.*, 1993, pp. 164–171.

[11] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.

[12] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient ConvNets," 2016, *arXiv:1608.08710*. [Online]. Available: http://arxiv.org/abs/1608.08710
[13] E. Fiesler, A. Choudry, and H. J. Caulfield, "Weight discretization paradigm for optical neural networks," in *Optical Interconnections and Networks*, vol. 1281. International Society for Optics and Photonics, 1990, pp. 164–174, doi: 10.1117/12.20700.
[14] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," 2014, *arXiv:1412.6115*. [Online]. Available: http://arxiv.org/abs/1412.6115
[15] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*. [Online]. Available: http://arxiv.org/abs/1510.00149
[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531
[17] 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.*, Jul. 2017, pp. 4133–4141.
[18] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 1921–1930.
[19] Q. Guo, Z. Yu, Y. Wu, D. Liang, H. Qin, and J. Yan, "Dynamic recursive neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5147–5156.
[20] D. Zhou, X. Jin, Q. Hou, K. Wang, J. Yang, and J. Feng, "Neural epitome search for architecture-agnostic network compression," in *Proc. Int. Conf. Learn. Represent.*, 2020. [Online]. Available: https://openreview.net/forum?id=HyxjOyrKvr
[21] Y. Yang, J. Yu, N. Jojic, J. Huan, and T. S. Huang, "FSNet: Compression of deep convolutional neural networks by filter summary," 2019, *arXiv:1902.03264*. [Online]. Available: http://arxiv.org/abs/1902.03264
[22] E. Shin and S.-H. Bae, "Global weight: Network level weight sharing for compression of deep neural network," in *Proc. Korean Soc. Broadcast Eng. Conf.* The Korean Institute of Broadcast and Media Engineers, 2020, pp. 22–25. [Online]. Available: http://www.koreascience.kr/article/CFKO202023758833858.page
[23] 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*. [Online]. Available: http://arxiv.org/abs/1704.04861
[24] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[26] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet v2: Practical guidelines for efficient cnn architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
[27] P. Stock, A. Joulin, R. Gribonval, B. Graham, and H. Jégou, "And the bit goes down: Revisiting the quantization of neural networks," 2019, *arXiv:1907.05686*. [Online]. Available: http://arxiv.org/abs/1907.05686
[28] O. Kopuklu, M. Babaee, S. Hörmann, and G. Rigoll, "Convolutional neural networks with layer reuse," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 345–349.
[29] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[30] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[31] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2736–2744.
[32] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "AMC: Automl for model compression and acceleration on mobile devices," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 784–800.
[33] D. Lin, S. Talathi, and S. Annapureddy, "Fixed point quantization of deep convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2849–2858.
[34] W. Sung, S. Shin, and K. Hwang, "Resiliency of deep neural networks under quantization," 2015, *arXiv:1511.06488*. [Online]. Available: http://arxiv.org/abs/1511.06488
[35] B. Zhuang, C. Shen, M. Tan, L. Liu, and I. Reid, "Towards effective low-bitwidth convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7920–7928.
[36] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, and Z. Zhang, "Improving neural network quantization without retraining using outlier channel splitting," 2019, *arXiv:1901.09504*. [Online]. Available: http://arxiv.org/abs/1901.09504
[37] M. Courbariaux, Y. Bengio, and J.-P. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3123–3131.
[38] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to $+1$ or -1," 2016, *arXiv:1602.02830*. [Online]. Available: http://arxiv.org/abs/1602.02830
[39] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6869–6898, 2017.
[40] D. Haase and M. Amthor, "Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved MobileNets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14600–14609.
[41] K. Rijkse, "H.263: Video coding for low-bit-rate communication," *IEEE Commun. Mag.*, vol. 34, no. 12, pp. 42–45, Dec. 1996.
[42] Y. Wang, C. Xu, S. You, D. Tao, and C. Xu, "CNNpack: Packing convolutional neural networks in the frequency domain," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 253–261.
[43] J. H. Ko, D. Kim, T. Na, J. Kung, and S. Mukhopadhyay, "Adaptive weight compression for memory-efficient neural networks," in *Proc. Conf. Design, Autom. Test Eur.* European Design and Automation Association, 2017, pp. 199–204. [Online]. Available: https://ieeexplore.ieee.org/document/7926982
[44] S. M. Ross, "Using statistics to summarize data sets," in *Introductory Statistics*, 3rd ed., S. M. Ross, Ed. Boston, MA, USA: Academic, 2010, ch. 3, pp. 71–143.
[45] H. J. Seltman, "Experimental design and analysis," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. 428, 2012.
[46] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2704–2713.
[47] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.

KANG-HO LEE received the B.S. degree from the Department of Electronic Engineering and the Department of Computer Science and Engineering, Kyung Hee University, South Korea, in February 2019, and the M.S. degree from the Department of Computer Science and Engineering, Kyung Hee University, in February 2021. His research interests include deep learning model compression and few-shot learning in the computer vision task.

SUNG-HO BAE (Member, IEEE) received the B.S. degree from Kyung Hee University, South Korea, in 2011, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012 and 2016, respectively. From 2016 to 2017, he was a Postdoctoral Associate with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), MA, USA. Since 2017, he has been an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University. He has been involved in model compression/interpretation for deep neural networks and inverse problems in image processing and computer vision.