IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 4, 2021, accepted April 24, 2021, date of publication May 5, 2021, date of current version May 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077597

Optimizing Spatial Shift Point-Wise Quantization

EUNHUI KIM, (Member, IEEE), KYONG-HA LEE, (Member, IEEE), AND WON-KYUNG SUNG
Korea Institute of Science and Technology Information, Daejeon 34131, South Korea

Corresponding author: Eunhui Kim (ehkim@kisti.re.kr)

This work was supported by the Korea Institute of Science and Technology Information under Grant K-21-L04-C01.

ABSTRACT It is no longer an option but a necessity to enhance the efficiency of deep learning models regarding energy consumption, learning time, and model size as the computational burden on deep neural networks increases. To improve the efficiency of deep learning, this study proposes a lightweight spatial shift point-wise quantization (L-SSPQ) model to construct a ResNet-like CNN model with significantly reduced accuracy degradation. L-SSPQ adds efficiency with the last linear layer weight reduction technology to SSPQ, which combines compact neural network design and quantization technology. To reduce weight and optimize performance, the learning time and system-required resources in the L-SSPQ are minimized. Accuracy could be improved with the warm-up interval and a step-size optimal value, both of which are hyper-parameters of the cosine learning rate. A two-stage optimization method that divides quantization learning into two steps is applied to further minimize loss. The size of the proposed L-SSPQ50 model is only 3.55 MB with an accuracy loss rate of 2.42%. This is just 3.56% of the size of ResNet50. In addition, the L-SSPQ50 score was 1.318 for information density, surpassing the SOTA models, including MobileNet V.2, MobileNet V.3, ReActNet-A, and FracBNN.

INDEX TERMS Compact neural network design, quantization, lightweight modeling, convolution neural network.

I. INTRODUCTION

The performance of deep learning models generally improves as the size of the input data, depth of the model (number of layers), and width of the model (number of channels) increase. In the area of computer vision, the current SOTA model, Res-Next-101, has 829 M (8.2 billion) parameters [1]. In the natural language processing field, the current SOTA model, GPT-3 is composed of 98 layers of transformers, and the total number of parameters reaches 175 B (175 billion) [2]. Considering the ResNet50 model's four-day learning period [3] with 25.6 M (25.6 million) parameters using two RTX-2080 GPUs, approximately 128 days of training will be required for the Res-Next-101 model. It takes approximately 355 years of training for the GPT-3 model using an Nvidia V100 GPU. Thus, it is a necessity, rather than an option, to increase the efficiency of deep learning models regarding energy consumption, learning duration, and model size because the computational requirements for deep neural networks are increasing [4]. Therefore, lightweight modeling techniques have been developed to address this problem.

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

Lightweight modeling technology can be divided along two axes: 1) quantization and compact neural network technology that can model from scratch and 2) pruning and knowledge distillation based on pre-trained model data. Considering the learning time and system resource requirements, neither pruning nor knowledge distillation was used in this study. Pruning sets the target to remove the connections between neurons through iterative learning until optimization while knowledge distillation requires pre-learning by the teacher model or learning simultaneously with the teacher model. The quantization and compact neural network design methods, which consume learning time and system resources relatively economically, were used for lightweight modeling in this study.

Spatial Shift Point-Wise Quantization (SSPQ) model [5] is a lightweight method that combines compact neural network design and quantization technology, replacing spatial operation with shift operation and applying quantization to the point-wise convolution. The SSPQ model applies lightweight technology to minimize the loss of accuracy and performance. Spatial convolution uses a shift operation with an inverse residual block structure, whereas point-wise convolution uses weights and activation functions that ensure

convergence by calculating at 1-bit and 4-bit quantization precision, respectively. In addition, after passing the quantized activation function, batch normalization is applied to minimize the instability of the mean and variance for each layer that arises from quantization.

The main contribution of the proposed L-SSPQ is as follows:

- The proposed L-SSPQ model reduces the fully connected layer parameters from 8 MB to 2 MB by omitting the last block of the SSPQ model and directly connecting it to the average pooling layer to reduce the number of computational channels and quantize the last active function.
- While reducing the weight and optimizing performance, the learning time and required system resources in L-SSPQ are minimized.
  - Based on the Nesterov optimizer, the accuracy is improved by learning the hyper-parameter warm-up interval and step-size optimization value of the cosine learning rate. In particular, for the Cifar100 dataset, the SSPQ34 model shows an improvement of 1.19% compared to the fixed-learning rate method.
  - To minimize the quantization loss, the method in the SSPQ model is used. In addition, a two-stage optimization that divides quantization learning is applied, but the accuracy is improved by 0.48% by adding only one epoch to reflect the final learning rate of the first learning session during the second training of L-SSPQ50 on the ImageNet dataset.
- The proposed L-SSPQ50 model shows an accuracy loss rate of 2.42% with only 3.55 MB, which is 3.56% of the number of ResNet50's parameters and with 36.28% fewer parameters is 0.26% better than the SSPQ50 model. In addition, the L-SSPQ50 score of 1.318 surpasses the SOTA models, including MobileNet V.2 [6], MobileNet V.3 [7], ReActNet-A [8], and FracBNN [9] models regarding information density.

This paper is organized as follows. In Section II, the main concepts of the compact neural network, quantization, and the optimized training of quantization neural networks are introduced. In Section III, the structure and limitations of the SSPQ model, the structure of the neural network block of the proposed L-SSPQ model and the optimized training of the quantized neural network method are explained. Section IV describes the experimental results, and Section VI concludes the paper.

II. RELATED WORK

This section summarizes recent studies on deep learning and the lightweight modeling technology to be achieved through this study, i.e., compact neural network design, quantization, and the learning method to optimize deep learning quantization modeling.

A. COMPACT NEURAL NETWORK DESIGN

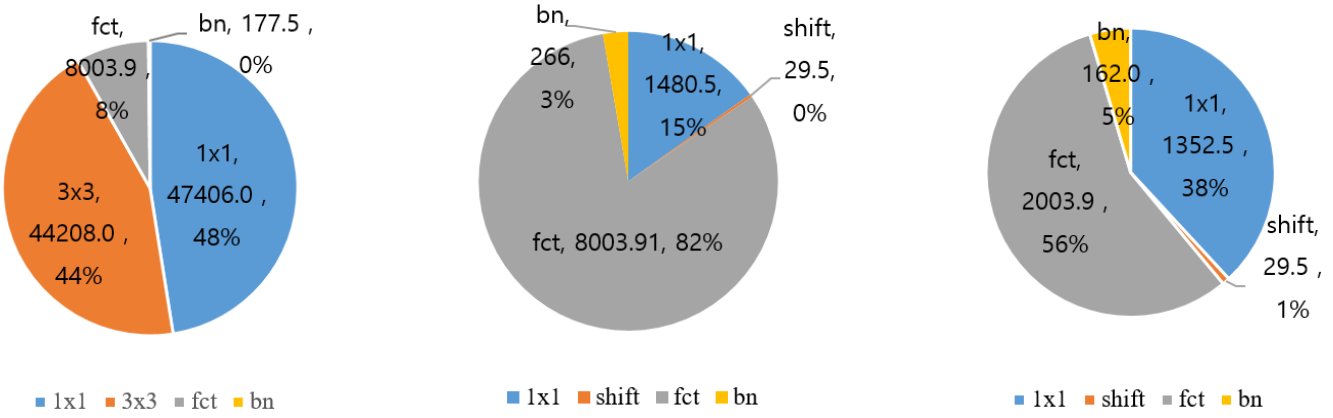
In 2015, ResNet [3] employed an identity connection for a convolution operation, in which an output map is a result of adding an input map and residual. The bottleneck block [10] is a structure in which the number of channels can be adjusted. Unlike general convolutional configuration block, the number of channels is reduced by adding each map in the direction of the channel through a 1×1 point-wise operation in the block. A convolution operation is then performed with this as the input. Subsequently, the number of channels is expanded through a 1×1 point-wise operation to match the number of input and output channels and form the same connection. The inverted residual block [6] structurally reduces the amount of computation and improves accuracy. For this, it connects the layers with a small number of channels and increases the number of channels of the convolution operation in the block instead of connecting the layers with a large number of channels in the identity connection.

By reinterpreting the general convolution operation, the depth-wise separable point-wise convolution shifts the multiplication operation calculated with the kernel for each channel by placing it in sequence (point-wise convolution operation) to set the order of operations. This has the advantage of reducing the amount of computation and enabling computational acceleration through parallel computation [11].

ShiftNet is another reinterpretation of convolution operations. The convolution operation result of a kernel whose one value is 1 and the remaining values are 0 among 3×3 kernels is equal to the result of shifting the input map into a specific direction. Instead of multiplication, the spatial convolution is replaced by a memory shift operation by dividing the channels into groups and randomly determining the direction of the shift [12]. To apply the gradient operation to the direction optimized for each channel movement, the active shift operation replaces the shift operation and adds sophistication by increasing the accuracy of the shift convolution operation [13]. Quantization of the channel movement is applied in the sparse shift operation to reduce the load of the memory shift operation through the channel shift operation [14].

B. QUANTIZATION

Quantization reduces the number of calculation bits, thus reducing the amount of calculation and the footprint size of the learning parameter, thereby improving the calculation speed. In deep learning lightweight modeling, quantization can be applied to weights, activation functions, and gradients. In particular, for binary quantization, the signum function (clip function) is used in the forward operation to convert the value above the reference value to 1 and convert the remaining value to -1 (or 0). In the backpropagation operation of binary quantization, gradient operation is impossible by differentiating the clip function. In this regard, by applying the straight-through-estimate method [15], which approximates the signum function using the derivative of the hard tanh



(a) ResNet50 (97.49 MB) (b) SSPQ50 (9.89 MB) (c) L-SSPQ50 (3.55 MB)

FIGURE 1. The ratio of learning parameters per layer of ResNet50, SSPQ50, and L-SSPQ50. The five types of layers are (1) [3 × 3] spatial conv. layer; (2) [shift] spatial conv. layer, (3) [1 × 1] point-wise conv. layer, (4) [fct] fully connected layer(linear layer), and (5) [bn] batch normalization layer. (the unit of the learning parameters in the figure is KB).

(hyperbolic tangent) function, BinaryNet [16] showed meaningful performance by quantizing the weights and activation functions. Quantization modeling incurs a loss of accuracy. In particular, among the binary quantization models, XNOR-Net [17] incurs 18.1% accuracy degradation in ImageNet dataset compared to ResNet18. As evidenced by recent analysis [18], when the number of quantization modeling bits falls below four, an error occurs, which makes it difficult to maintain robustness while quantizing. As the difference increases, stable convergence cannot be achieved, resulting in a significant loss of accuracy.

C. OPTIMIZED LEARNING OF QUANTIZED NEURAL NETWORKS

Several learning methods to improve the quantization loss have been studied. Zhuang *et al.* verified that the 4-bit quantization model is more competitive than the full-precision model by applying three methods: two-stage optimization (TS), progressive quantization (PQ), and transfer-learning (guided) [19]. Recently, Martinez *et al.* confirmed improved performance in binary models similarly by sequential learning method from full precision state to binary bit state. [20]. TS is learned by performing only weight quantization and then simultaneously learning while performing active function quantization using the learned weight values as initial values. The quantization model that mixes the three methods, TS, PQ, and guided, outperforms the full-precision model. However, compared to the general model, the training time is 2× for TS, 4× for PQ, and 3× for guided. In addition, the guided method is subject to a restriction that requires more than twice the system resources simultaneously. Therefore, in this study, only the TS method with a low learning time and system requirements was applied, but when the trained parameters and the final learning rate of the first learning were used for the second learning, the performance was improved with only one epoch of learning. Recently, ReActNet [8] and FracNet [9] showed improved performance even in binary networks by applying the learning method of Martinez [20] with the modified activation function to achieve the convergence of this data distribution.

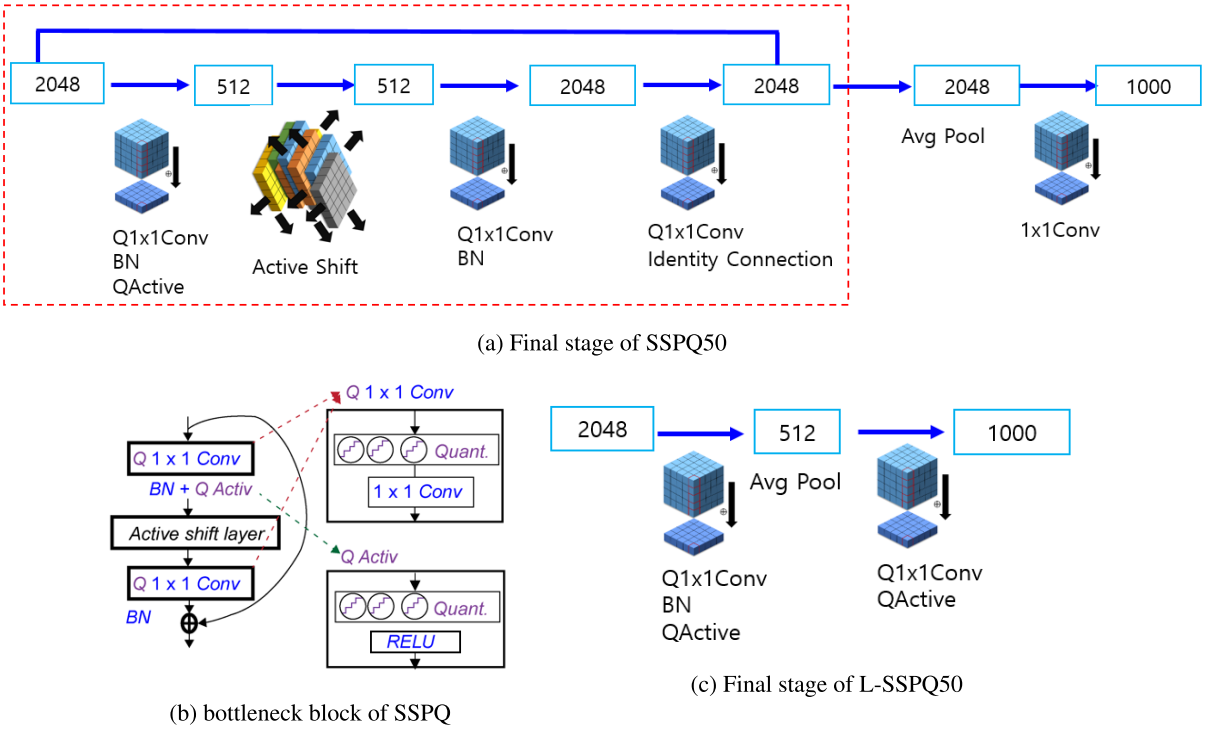
This study differs from the above ones since it enhances performance using compact neural network design technology, quantization, and its learning method improvement concurrently. This paper proposes a two-stage learning method that reduces quantization errors while adjusting the variation in the learning rate to the warm-up length and step size with low precision. In addition, the proposed model achieved superior performance to the above models based on the information density metric.

III. OPTIMIZING LIGHTWEIGHT SPATIAL-SHIFT POINT-WISE QUANTIZATION MODEL

In this section, we describe the lightweight technology and optimization technology of the proposed L-SSPQ model. First, the structure and limitations of the SSPQ model, the predecessor of the L-SSPQ model, are examined.

A. LIMITATIONS OF SSPQ MODEL

Figure 1 shows the ratio of learning parameters per layer of the ResNet50 and lightweight models. Figure 1(b) shows the characteristics of each layer compressed via the SSPQ model. The fully connected layer, i.e., the last linear layer, has much room for additional compression. Figure 1(c) shows the size of the learning parameters for each layer of the L-SSPQ model presented in this study. Through additional lightening of the linear layer, L-SSPQ confirms that the accuracy classification performance improved by 0.26% compared to that of the SSPQ model for the ImageNet dataset with a size of 36.28% of the SSPQ model. That is, for the ImageNet dataset, the L-SSPQ50 model confirms an accuracy of 73.92%, which is improved by 0.26% to the 3.55 MB



(a) Final stage of SSPQ50

(b) bottleneck block of SSPQ

(c) Final stage of L-SSPQ50

FIGURE 2. Light-weighting the last stage of SSPQ model. (a) and (b) depicts the final stage and the bottleneck block of SSPQ model, respectively. (c) describes the final stage of the L-SSPQ which omits the last bottleneck block and applys quantization to the last linear layer.

learning parameter size, which is 36.28% the size of the SSPQ50 model and 3.56% the size of the ResNet50 model.

B. STRUCTURE OF THE SSPQ MODEL

The SSPQ model [5] uses a lightweight method that combines a compact neural network design and quantization technology. As a method of applying a lightweight technique but minimizing the loss of accuracy and performance, the basic block of the SSPQ model is composed of similar inverted residual bottleneck blocks [6]. As shown in the red square frame in Figure 2(a) and Figure 2(b), the 3×3 convolutional function of the two layers was replaced by one shift layer [13]. Only the 1×1 point-wise convolution part was quantized. SSPQ uses Dorefa-Net [21] for quantization, which supports multi-bit quantization and has the advantage of flexibility in obtaining quantization derivatives as it is composed of a differentiable hyperbolic-tangent equation. The quantization size is set one bit for weights and four bits for the activation function for which convergence is guaranteed [18]. After passing through the quantized activation function, batch normalization is applied to minimize the instability of each mean and variance layer caused by quantization.

C. LIGHTWEIGHT SPATIAL-SHIFT POINT-WISE QUANTIZATION MODEL

As explained in Section III-A, SSPQ has room to reduce the last linear layer. In L-SSPQ, to reduce the last linear layer, we adopt the transformation of the linear layer proposed in MobileNet V.3 [7]. The model is further reduced by applying quantization. Through this, the proposed L-SSPQ confirms that the accuracy improves by 6.47% using only 38.17% of the learning parameter size compared to the MobileNet v.3 small model.

The basic block of the SSPQ model is composed of similar inverted residual bottleneck blocks [6]. In all layers, the number of channels of each block is [64, 128, 256, 512], and the number of block hierarchies of the corresponding channel number is [3, 4, 6, 3], and this doubles the channel for each block. In addition, it is a bottleneck block structure that expands the channel four times within the block. The output of the block immediately before the last linear layer (fully connected layer) becomes an output map of 2,048 pixels with 512 channels expanded by a factor of four. The last block and linear layer of the SSPQ are shown in Figure 2(a), and the final stage of the L-SSPQ with the lighter weight is depicted in Figure 2(c).

As shown in Figure 2, in the L-SSPQ, the last block that consists of four layers is omitted. The 2,048-output channel, which is the output of the previous block, is received as an input, and a 1×1 convolution is calculated based on the low-precision weight that has passed through a one-bit quantization function. After normalization, the result of the ReLU activation function passing through the four-bit quantization function becomes the number of the 512 channels, which is 1/4 the number of the 2,048 channels passed to the previous average integration process. After the average pooling

process using these 512 channels as the input, after passing the activation function again, it performs four-bit quantization and connects to the fully connected layer, ImageNet 1,000 classes. In L-SSPQ, similar to SSPQ, the quantization function uses DorefaNet [21].

By transforming the last layer of SSPQ and applying additional quantization to the fully connected layer, L-SSPQ has been reduced to 36.38% of the SSPQ model training parameter size. In general, a reduction in the size of a learning parameter causes a loss of accuracy. Therefore, in the next section, we explore a training method optimized for quantized neural networks.

D. OPTIMIZING LIGHTWEIGHT SPATIAL-SHIFT POINT-WISE QUANTIZATION MODEL

The optimized result of the change in the learning rate, in the form of a half-cycle of the cosine function over the entire learning period without periodically repeating the change in the learning rate, can be obtained from the experimental results in Loshchilov's study [22]. In general, the optimal learning rate must be determined through various learning rate changes. Conversely, the cosine learning rate has the advantage of finding the optimal condition because it adjusts the learning rate, where the learning rate initially is set to a large value and then reduced to a small value when learning reaches convergence. In this study, the cosine learning rate is optimized using the Nesterov optimizer. During the range where epoch e is the warm-up period T_w, i.e., $(e_t < T_w)$, $\eta_t = \eta_{max}(= 0.1)$ is set. When the step size S is reached, η_t is periodically updated using Equation (1). The minimum value of η_t is set as η_{min} to prevent the problem of not being updated because it is too small.

$$\eta_t = \eta_{t-1} \cdot (\eta_{min} + \tfrac{1}{2}(1 - \eta_{min})(1 + cos(\tfrac{e_t - T_w}{T_m - T_w}\pi))) \tag{1}$$

In Equation (1), T_m is the maximum period (max epoch), T_w is the warm-up epoch length, and e_t is the epoch of the current t_{th} learning. To find the optimal learning condition, we conducted a comparative experiment based on the difference in learning accuracy according to the warm-up length T_w and step size S conditions. Figure 3 provides an example of a change in the learning rate according to T_w, S and T_m.

As verified by the research of Zhuang *et al.* [19] and Martinez *et al.* [20], a complete, well-learned method of reducing the loss through quantization and obtaining the performance before quantization is introduced in Section II-C. After only quantizing the weights, quantization is performed on both the weight and activation function based on the trained parameters, as described in Algorithm 1. Unlike the algorithm proposed by Zhuang *et al.* [19], after learning for all epochs in the first step (quantization for weights only) of TS, the trained parameters and hyper-parameters (learning rate) of the first learning are used to learn an additional T_{s2} epoch. The number of T_{s2} epochs is affected by the learning rate, η_{min}. In the case of 1e-6, where the learning rate is low and the additional learning content is hardly reflected, one epoch is sufficient. In the case of 1e-3, where the learning rate affects the change in learning content, it converges to 20 epochs.

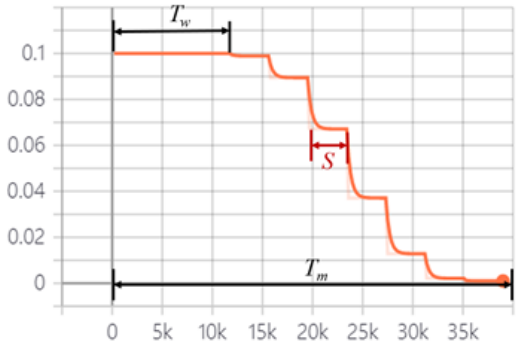


FIGURE 3. Example of change in learning rate during warmup T_w and step size S in the whole epoch T_m.

Algorithm 1 Two-Stage Quantization Algorithm

Input: Training data (x_i, y_i); full-precision L-SSPQ model
Output: low-precision L-SSPQ model with weights W as q_w-bit and activations A as q_a-bit (Wq_w, Aq_a)
Stage1, Quantize Wq_w

```
for e = 1, .., T_m do
    if e%S == 0 then
        η_t update by Equation (1)
    end if
    for t = 1, .., L do
        Randomly sample mini-batch data; learning L-SSPQ with Wq_w and A32 precision states
    end for
end for
```

Stage2, Quantize Aq_a and Wq_w

```
Initialize W, A and learning rate η as Stage1 results
for e = 1, . . . , T_s2 do
    for t = 1, . . . , L do
        learning L-SSPQ with Wq_w and Aq_a precision states
    end for
end for
```

IV. EXPERIMENTAL RESULTS

A. EXPERIMENT ENVIRONMENT

In this study, we configured the hardware of an Intel Core i7 computer with NVME SSD, 64 GB RAM, and two RTX 2080 GPUs (the GPU RAM was 24 GB each, 48 GB in total). Tensorflow 1.15.3 version was installed as the source level on a Linux Ubuntu 16.04 OS. For the experimental data set, Cifar10, Cifar100, and ILSVRC2012-ImageNet datasets were used with classes of 10, 100, and 1,000, respectively, which are widely utilized for image classification performance evaluation. Table 1 provides a detailed description of each dataset. In the Cifar10 and Cifar100 datasets, the

TABLE 1. Three datasets for evaluating the L-SSPQ.

dataset	# of train	# of test	# per class	image size
Cifar10	50,000	10,000	6,000	32x32
Cifar100	50,000	10,000	600	32x32
ImageNet	1,200,000	50,000	1,250	256x256

average value of the three experiments was used to increase the reliability of the performance for each hyper-parameter experiment. The max epoch T_m in Algorithm 1 was set to 105 in the ImageNet dataset and 200 in the Cifar10 and Cifar100 datasets. The batch size was set to 256 in the ImageNet and Cifar100 datasets and 128 in the Cifar10 dataset.

For the Cifar10 dataset, the proposed L-SSPQ model calculates the weights and activation functions with 4-bit precision for the depth 20 (L20-W4A4) and the depth 20-wide (L20-wide-W4A4) model that expanded the channel. The channel size for each depth 20 layer (L20) is [16, 32, 64,128], and the number of allocated blocks for each channel size is [2, 2, 3, 2]. For the wide model (L20-wide) by increasing the number of channels, as proposed in [23], the number of channels per block and the number of allocated blocks are [64, 128, 256] and [3, 3, 3], respectively.

For the Cifar100 dataset, the proposed L-SSPQ model calculates the weights and activation functions with 8-bit precision for the depth 34 (L34-W8A8) and depth28-wide (L28-wide-W8A8) model that expanded the channel. The channel size for each depth34 layer (L34) is [16, 32, 64, 128], and the number of allocated blocks for each channel size is [3, 4, 6, 3]. For the wide model (L28-wide) by increasing the number of channels, as proposed in [23], the number of channels per block and the number of allocated blocks are [64, 128, 256] and [4, 6, 3], respectively.

For the ImageNet dataset, the channel size for each depth50 layer (L50) is [64, 128, 256, 512], and the number of allocated blocks for each channel size is [3, 4, 6, 3]. The inverted residual bottleneck blocks are used, as explained in Section III-C.

B. PERFORMANCE COMPARISON OF L-SSPQ WITH OTHER MODELS IN IMAGENET DATASET

As shown in Figure 2, the SSPQ model is additionally compressed by transforming the linear layer. Therefore, as shown in Figure 1, the fully connected layer (linear layer), which has 82% of the learning parameters of the SSPQ50 with a size of 8 MB, decreases to 56% of the total parameters from L-SSPQ50 to 2 MB. In addition, L-SSPQ applies the optimization technique of the algorithm 1 learning method. As shown in Table 2, the L-SSPQ model, which incorporates lightweight technology and learning optimization technology, demonstrates an accuracy of 73.92% in the ImageNet dataset.

Information density is a method for measuring how well information is condensed and utilized by observing the accuracy rate versus the size of the learning parameter,

TABLE 2. Performance comparison among the SSPQ50 v.2 and other comparative models in the ImageNet dataset.

Model	Size of parameters	top1	I.D.
ResNet50 [10]	25.5M (97.49MB)	**76.34**%	-0.106
ASL50 [13]	22.21M (84.74MB)	72.28%	-0.069
ResIBShift50 [14]	14.26M(54.41MB)	75.83%	0.144
MobileNet v.2 [6]	3.50M(13.04MB)	72.2%	0.743
SSPQ50 [5]	14.26M(9.78MB)	73.66%	0.877
Mobilenet v.3 large [7]	5.4M(20.09MB)	75.2%	0.573
Mobilenet v.3 small [7]	2.5M(9.3MB)	67.4%	0.86
ReActNet-A [8]	4.56MB	69.4%	1.146
FracBNN [9]	4.56MB	71.8%	1.197
SSPQ50 v.2	11.64M(**3.55MB**)	73.92%	**1.318**

TABLE 3. Performance comparison between SSPQ and L-SSPQ according to dataset.

Data	model	SSPQ		L-SSPQ flr	
		Size	Accuracy	Size	Accuracy
C10	L20-W4A4	0.34 MB	88.71%	0.32 MB	87.99%
	L20wide-W4A4	1.51 MB	91.86%	1.25 MB	91.11%
C100	L34-W8A8	0.64 MB	63.70%	0.59 MB	63.53%
	L28wide-W8A8	0.82 MB	68.97%	0.74 MB	69.45%
I1000	L50-W1A4	9.78 MB	73.66%	3.55 MB	73.39%

* flr is the fixed learning rate.
* L20 denotes the number of layers.
* wide means channel expansion as explained in [23]
* W1 and A4 indicate that the number of quantization bits used is 1 for the weight W and 4 for activation.

as expressed by Equation (2) [24].

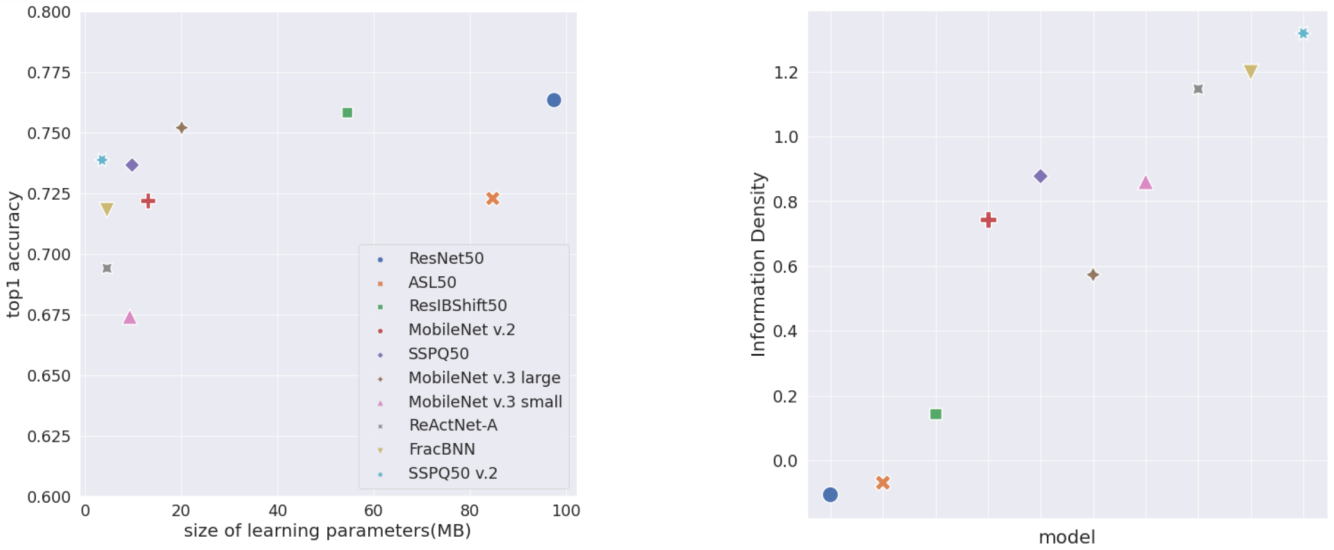
$$ID(m) = log\frac{Accuracy(m)}{Size\ of\ Parameters(m)} \tag{2}$$

The information density metric reveals that the point in the upper right corner of Figure 4(b) is the most efficient condensation of information per bit. As shown in Figure 4(b), the SSPQ50 v.2 (L-SSPQ50) model shows the best performance in terms of information density compared to MobileNet V.2 [6], MobileNet V.3 [7], ReActNet-A [8], and FracBNN [9].

C. COMPARISON OF L-SSPQ MODEL PERFORMANCE CHANGES ACCORDING TO OPTIMIZATION

Table 3 shows the model size and learning accuracy of the SSPQ and L-SSPQ models. The L-SSPQ flr is the result of learning by applying a fixed learning rate to the learning rate change in the same manner as is SSPQ. That is, for the Cifar10(C10) and Cifar100(C100) datasets, training was performed with the momentum optimizer with a value of [0.1, 0.01, 0.001] for the epoch [0, 100, 150] with a fixed learning rate. For the ImageNet(I1000) dataset, the epoch [0, 30, 60, 90, 100] for each of the results of Nesterov learning was compared by setting the learning rate to a value of $\times 0.1$, starting at 0.1. Thus, η_{min} is set as 1e-3 and 1e-6 for the Cifar and ImageNet datasets, respectively. Therefore, for the step2 epoch T_{s2}, 1 and 20 epochs were used in the experiment for the ImageNet and Cifar datasets, respectively.

As explained in Table 3, the L-SSPQ model reduces the number of training parameters compared to the SSPQ model, and thus the accuracy performance decreases.



(a) Accuracy according to the size of parameters

(b) Information density

FIGURE 4. Performance comparison between the proposed model L-SSPQ50 and lightweight models, and it can be confirmed through the figure on the right that the L-SSPQ50 shows the best performance in terms of information density.

TABLE 4. Performance comparison of L-SSPQ between flr and clr according to dataset.

Data	model	SSPQ	LSSPQ flr	LSSPQ clr(T_w, S)
C10	L20-W4A4	88.71%	87.99%	88.97%(5,30)
	L20wide-W4A4	91.86%	90.83%	91.11%(30,30)
C100	L34-W8A8	63.70%	63.53%	64.72%(30,30)
	L28wide-W8A8	68.97%	69.45%	70.49%(30,30)
I1000	L50-W1A4	73.66%	73.39%	73.92%(5,10)

* flr is the fixed learning rate, and clr is the cosine learning rate.

Accordingly, Table 4 shows the results of the experiment of the SSPQ, L-SSPQ flr, and L-SSPQ clr by finding the optimized warm-up T_w length and step size S of the cosine learning rate.

In particular, Table 4 presents the experimental results of combining the Nesterov algorithm and the cosine learning rate and applying the two-stage algorithm of Algorithm 1 and applying sequential quantization. As shown in Table 4, the experimental results using L-SSPQ clr improved accuracy from at least 0.28% to at most 1.19% compared to L-SSPQ flr. Compared to the SSPQ model, the L-SSPQ clr model has improved accuracy except for the 20 layer wide model of the cifar10 dataset. In particular, in the case of the 28 layer wide model for the cifar100 dataset, the accuracy performance was improved by 1.52%.

V. DISCUSSION

In this paper, we can achieve the best performance in the metric of information density by reducing the size of learning parameters combining compact neural network technology and quantization method and by improving the accuracy using optimized training techniques for quantization neural networks.

A. DIFFERENCE BETWEEN SSPQ AND L-SSPQ FOR COMPACT NEURAL NETWORK DESIGN

In SSPQ, to reduce the number of learning parameters, the spatial convolution of the inverted residual block was replaced by shift convolution and the quantization operation was applied to the point-wise convolution. In addition, SSPQ minimizes the loss of accuracy while reducing the size of the quantization model. The quantization bit size of the activation function and the quantization bit size of the weight were set to 4bit and 1bit, respectively for both of accuracy and model size. When the number of quantization bits decreases, the point-wise convolution is relatively robust to the problem of unstable convergence during learning compared to spatial convolution, and quantization is applied to point-wise convolution except shift convolution. [18]. For quantization, we adopt Dorefa-Net, which supports a multi-bit quantization function which mapped a differentiable hyperbolic-tangent function [21].

The layers of the SSPQ model were compared in terms of the learning parameter compositions, revealing that the last linear layer had more room for further learning parameter size reduction, as shown in Figure 1.

In L-SSPQ, to additionally reduce learning parameter, the last layer is transformed by omitting the last bottleneck block and applying quantization to the last linear layer and 6MB can be decreased as described in Figure 1 and Figure 2. In order to improve the accuracy even when the learning parameter is reduced, L-SSPQ applied a two-step learning strategy optimized for quantization, and presented a method to find the optimized performance according to the hyper-parameter T_w and S of the cosine learning rate as Equation (1).

B. FINDINGS IN OPTIMIZED TWO-STAGE LEARNING METHOD FOR QUANTIZED NEURAL NETWORK

Zhuang's research [19] is a technique for optimizing the quantization model, and by combining the three methods TS, PQ, and Guided, we confirmed the improved performance than the full-precision model. However, the TS, PQ, and Guided methods have a limitation that increases the learning time by more than 4 times. This study applied the Two Stage (TS) strategy, but found a way to further save learning time. It reduces the learning epoch when the first stage of learning is completed and the second stage of learning commences. When the size of the learning rate reaches 1e-6, there is almost no learning update, so an additional 1 epoch can be used to obtain optimized performance. Since the size of 1e-3 has a training update, it was found that approximately 20 epochs can be additionally trained to obtain optimized performance.

C. APPLICATION AND LIMITATION OF THIS STUDY

Recently, various IoT devices, including drones and robots, are attempting to apply deep learning models that process video, voice, and natural language for accurate movement and judgment under the limited computing power and memory of the embedded environment. In such a limited environment, the weight reduction of the enormous deep learning model is not an option, but a necessity. In this study, the size of the model is estimated by assuming the quantization bit. Therefore, the limitations of this study are limited in discussing the exact weight reduction size and inference speed. When a lightweight model is uploaded directly in the FPGA environment, the difference of the weight reduction size and the inference speed can be discussed.

VI. CONCLUSION

The L-SSPQ model combines compact neural network technology, quantization technology used in the SSPQ model, lightweight technology, and optimization technology. The experiment shows that the L-SSPQ50 model achieves an accuracy of 73.92%, which is improved by 0.26% when using 36.28% of the total training parameter size compared to the SSPQ50 model. Using a learning parameter size of 3.56% compared to the ResNet50 model, we confirmed an accuracy loss of 2.42%. The expected future work related to this study is the speed optimization of cuda-based novel shift convolution including stride operation. In the future, CPU and GPU operators that support four-bit quantization models will be standardized, and binary quantization models are expected to be supported in the next five years. It is expected that the role of lightweight deep-learning models will increase significantly through the availability of hardware supporting quantization models.

REFERENCES

[1] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

[2] T. B. Brown *et al.*, "Language models are few-shot learners," in *Proc. 34th Conf. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2020.

[3] 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.

[4] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017, *arXiv:1710.09282*. [Online]. Available: http://arxiv.org/abs/1710.09282

[5] E. Kim and K.-H. Lee, "Spatial shift point-wise quantization," *IEEE Access*, vol. 8, pp. 207683–207690, 2020.

[6] 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.

[7] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[8] Z. Liu, Z. Shen, M. Savvides, and K.-T. Cheng, "ReActNet: Towards precise binary neural network with generalized activation functions," 2020, *arXiv:2003.03488*. [Online]. Available: http://arxiv.org/abs/2003.03488

[9] Y. Zhang, J. Pan, X. Liu, H. Chen, D. Chen, and Z. Zhang, "FracBNN: Accurate and FPGA-efficient binary neural networks with fractional activations," 2020, *arXiv:2012.12206*. [Online]. Available: http://arxiv.org/abs/2012.12206

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 630–645.

[11] 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

[12] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao, N. Golmant, A. Gholaminejad, J. Gonzalez, and K. Keutzer, "Shift: A zero FLOP, zero parameter alternative to spatial convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9127–9135.

[13] Y. Jeon and J. Kim, "Constructing fast network through deconstruction of convolution," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5951–5961.

[14] W. Chen, D. Xie, Y. Zhang, and S. Pu, "All you need is a few shifts: Designing efficient convolutional neural networks for image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7241–7250.

[15] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*. [Online]. Available: http://arxiv.org/abs/1308.3432

[16] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 4107–4115.

[17] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 525–542.

[18] E. Park and S. Yoo, "Profit: A novel training method for sub-4-bit mobilenet models," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 430–446.

[19] 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.

[20] B. Martinez, J. Yang, A. Bulat, and G. Tzimiropoulos, "Training binary neural networks with real-to-binary convolutions," 2020, *arXiv:2003.11535*. [Online]. Available: http://arxiv.org/abs/2003.11535

[21] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients," 2016, *arXiv:1606.06160*. [Online]. Available: http://arxiv.org/abs/1606.06160

[22] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–16.

[23] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*. [Online]. Available: http://arxiv.org/abs/1605.07146

[24] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, "SqueezeNext: Hardware-aware neural network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1638–1647.

EUNHUI KIM (Member, IEEE) received the B.S. degree in information communication engineering from Chungnam National University, South Korea, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2009 and 2015, respectively. From 2000 to 2007, she was a Researcher with Samsung Electronics, Seoul, South Korea. From 2015 to 2016, she was a Postdoctoral Researcher with the Information Electronics Research Institute, KAIST. From 2017 to 2018, she was a Postdoctoral Researcher with the Cho Chun Shik Graduate School of Green Transportation, KAIST. In 2018, she was an Invited Professor with the National Center of Excellence in Software, Chungnam National University. Since 2019, she has been worked with the Korea Institute of Science and Technology Information as a Postdoctoral Researcher. Her research interests include machine learning, recommendation systems, intelligent transportation, and lightweight deep neural networks modeling in vision and language processing.

KYONG-HA LEE (Member, IEEE) received the B.S. and M.S. degrees in information and telecommunications engineering and the Ph.D. degree in computer engineering from Chungnam National University, South Korea, in 1998, 2000, and 2006, respectively. From 2007 to 2008, he was a Researcher with the Software Research Center. From 2008 to 2010, he was a Postdoctoral Researcher with the Computer Science Department, University of Arizona, USA. From 2010 to 2013, he was a Research Professor with the Computer Science Department, KAIST, South Korea. Since 2014, he has been a Senior Researcher with the Korea Institute of Science and Technology and Information. He has been an Associate Professor with the University of Science and Technology, South Korea, since 2020. Since 2018, he has been served as a Board Member for the Database Society, Korea Institute of Information Science and Engineering (KIISE) and an Associate Editor of the communications with KIISE, in 2020. His research interests include deep learning SW, scalable computing, and database systems.

WON-KYUNG SUNG received the Ph.D. degree in computational linguistics from Université Denis Diderot (Paris VII), in 1996. From 1998 to 2001, he was the Director of the SLP Division, L&H Korea. From 2001 to 2003, he was the Director of the Research and Development Center, Voicetech. From 2004 to 2018, he was the Director-General of the Convergence Technology Research Division, Korea Institute of Science and Technology Information (KISTI). Since 2013, he has been a Civilian Member, Ministry of the Interior. From 2013 to 2016, he was a Member of Council, Ministry of Culture, Sports and Tourism. Since 2017, he has been a Committee Member with the Ministry of Science and ICT. Since 2018, he has been a Professor with the Department of Data and HPC Science, University of Science and Technology (UST). He has been the Director with the Intelligent Infrastructure Technology Research Center, Korea Institute of Science and Technology Information (KISTI), since 2019. His research interests include machine learning, big data, digital twins, and language processing.