Lightweight Maize Disease Detection through Post-Training Quantization with Similarity Preservation

Carlos Victorino Padeiro Tse-Wei Chen Takahiro Komamizu Ichiro Ide
Nagoya University
Nagoya, Aichi 464-8601 Japan
padeiroc@cs.is.i.nagoya-u.ac.jp, twchen@ieee.org , taka-coma@acm.org, ide@i.nagoya-u.ac.jp

Abstract

Traditional crop disease diagnosis, reliant on expert visual observation, is expensive, time-consuming, and prone to error. While Convolutional Neural Networks (CNNs) offer promising alternatives, their high resource demands limit their accessibility to farmers, particularly those in resource-constrained settings. Lightweight models that operate on resource-limited devices without network access are crucial to address this gap. This paper proposes a Similarity-Preserving Quantization (SPQ) method to convert high-precision CNNs into lower-precision models while maintaining similar feature representations. While quantization offers a promising approach for building lightweight CNNs for crop disease detection, the quality of quantized models often suffers. SPQ addresses this challenge by ensuring equivalent activation patterns for similar crop images in both the original and quantized models. Experimental evaluation using MobileNetV2 and ResNet-50 demonstrates that SPQ improves throughput, inference, and memory footprint more than 3 times while preserving the detection performance.

1. Introduction

Crop diseases pose a formidable challenge to global food security, causing substantial production and economic losses, estimated at $220 billion annually by the United Nations Food and Agriculture Organization (FAO) [5]. These diseases contribute to insufficient human food supply and disrupt farmer's income-generating activities. Maize, one of the dominant food crops, is particularly susceptible to various diseases despite its global yield of $1.2 billion in 2020, with a productivity of 6.0 t/ha [6]. Conventional crop disease diagnosis methods rely on visual inspection by experts, utilizing their in-depth knowledge of crop diseases and their symptoms. This process is time-consuming, expensive, and prone to human error due to subjective perception. The advent of Convolutional Neural Networks (CNNs), particularly in image processing techniques, has revolutionized precision agriculture, labor costs, and high accuracy [22]. Previous studies, such as that by Zhang et al. [35], have proposed improved deep CNNs for crop disease detection. Besides, these models are computationally expensive and require significant memory due to over-parameterization, a common characteristic of deep neural networks [3]. This imposes high computational and memory demands for inference, making these solutions less accessible to farmers. Moreover, to address the network connectivity issues in remote cultivation areas, deploying these models on resource-constrained devices is essential for broader adoption. To address these challenges, we develop a lightweight object detection model designed explicitly to detect crop diseases.

This paper applies Post-Training Quantization (PTQ) [9] to reduce a neural network model's memory and computational requirements. PTQ is widely regarded as one of the most efficient compression methods in practice, benefitting from its data privacy and low computational costs. Emerging as a promising solution to resource limitations, it enables deploying resource-efficient CNNs for crop disease detection. Unlike conventional quantization requiring extensive calibration data and retraining, PTQ minimizes computational overhead by bypassing iterative fine-tuning. This efficiency gain, however, may lead to a minor trade-off in accuracy compared to full-precision models.

Recent research efforts have addressed this trade-off between accuracy and efficiency in PTQ. For instance, Nagel et al. [18] introduced soft quantization with learnable parameters by constructing new optimization functions based on second-order Taylor expansions of the loss functions before and after quantization. This approach effectively balances model accuracy and efficiency. Li et al. [13] proposed a block-by-block reconstruction method instead of the traditional layer-by-layer approach and utilized diagonal Fisher matrices to approximate the Hessian matrix, conserving more information during quantization. This strategy further improved quantization accuracy without compromising efficiency. Wei et al. [33] discovered that randomly disabling a subset of quantized activation feature map can smooth the loss surface of the quantization weights, leading to improved accuracy. Though PTQ has been studied and improved, these methods have yet to consider pairwise activation similarities between the full-precision and quantized models. Considering this, this paper proposes a quantization reconstruction method called SPQ (Similarity-Preserving Quantization) that preserves pairwise activation similarities between input pairs in the quantized model rather than directly mimicking internal representation space of the full-precision model. In summary, our main contributions are three-fold:

1. **SPQ**: We propose a PTQ method focusing on preserving pairwise activation similarities between input pairs in the quantized and full-precision models.
2. **Loss Function**: We apply a reconstruction error for preserving layer/block similarity between quantized and full-precision models to object detection neural network architectures.
3. **Validation**: We validate that SPQ enhances quantized network calibration outcomes and offers a valuable adjunct to established PTQ techniques.

2. Related Works

This section analyzes combinations of quantization techniques for enhancing resource-constrained disease detection applications.

2.1. Crop Disease Detection

Current techniques in precision agriculture are commonly preceded by analyzing the images captured by devices and sensors. These images are then used to detect crop diseases. The problem of crop disease detection has been addressed in numerous studies. However, most of them focused only on the problem of classifying foliar diseases, such as the method by *k*-means clustering and deep learning to detect orange diseases and predict their names from image [10]. Their architecture is designed based on GoogLeNet's [26] inception networks and AlexNet [12] for identifying and recognizing apple leaf diseases as proposed by Liu et al. [15]. A related work proposed a method based on an improved VGG-16 [25] network to identify apple leaf diseases [34]. Unlike these works, we apply object detection networks to detect crop diseases on plant leaves.

2.2. General Framework of Quantization

Quantization of neural networks has been studied for a while, and there are numerous methods [9, 13, 17, 33]. All of them are based on the following equations:

$$\begin{aligned} w_q &= \text{quant}(w), & (1)\\ \text{quant}(w) &= \text{clamp}\left(\left\lfloor \frac{w}{S} \right\rceil + Z; 0, 2^n - 1\right), & (2)\\ S &= \frac{w_{\text{max}} - w_{\text{min}}}{2^n - 1}, & (3) \end{aligned}$$

where S denotes the scaling factor to convert the range of w to n bit-width, and $\lfloor . \rceil$ is the round-to-nearest operation. The clamp function in Eq. 2 restricts a given value between an upper and lower bound.w_q is the quantized weight, and Z is used to decide which quantized value 0 is mapped to.

In contrast, the dequantization integer value represents $w' \in \mathbb{R}$ obtained by:

$$w' = w_q(S - Z). \quad (4)$$

The quantization procedure of the activation feature map is analogous to the weight value quantization, except that the minimum and maximum values are determined by analyzing activations from a limited calibration dataset and employing a moving average.

2.3. Post-Training Quantization

Quantization is a powerful technique for compressing neural networks, enabling their deployment on resource-constrained devices by applying Eq. 2. Two primary quantization methodologies exist: Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). QAT [4, 9, 24] incorporates quantization into the training phase, while PTQ [19] applies quantization after training completion.

PTQ offers significant computational advantages, making it the preferred choice for network deployment. The primary objective of PTQ is to determine the quantization parameters for weights and activations in each layer. Despite incorporating fine-tuning during quantization, these PTQ methods remain distinct from QAT. QAT employs the entire labeled training dataset to adjust the model's weights, while PTQ solely optimizes the quantization parameters using a subset of unlabeled data, making it efficient. Meanwhile, PTQ has emerged as a promising solution to address these challenges of lack of labeled data, enabling the deployment of CNN with significantly reduced memory footprint and computational complexity. Traditional quantization methods, such as full-precision training followed by quantization, often require large amounts of calibration data to fine-tune the quantized model, resulting in substantial computational overhead. In contrast, PTQ eliminates the need for iterative quantization training, significantly reducing computational costs and enabling efficient model deployment. However, this efficiency often comes at the partial sacrifice of accuracy due to the reduction in precision, which can lead to information loss and a diminished ability to represent fine-grained details in the model.

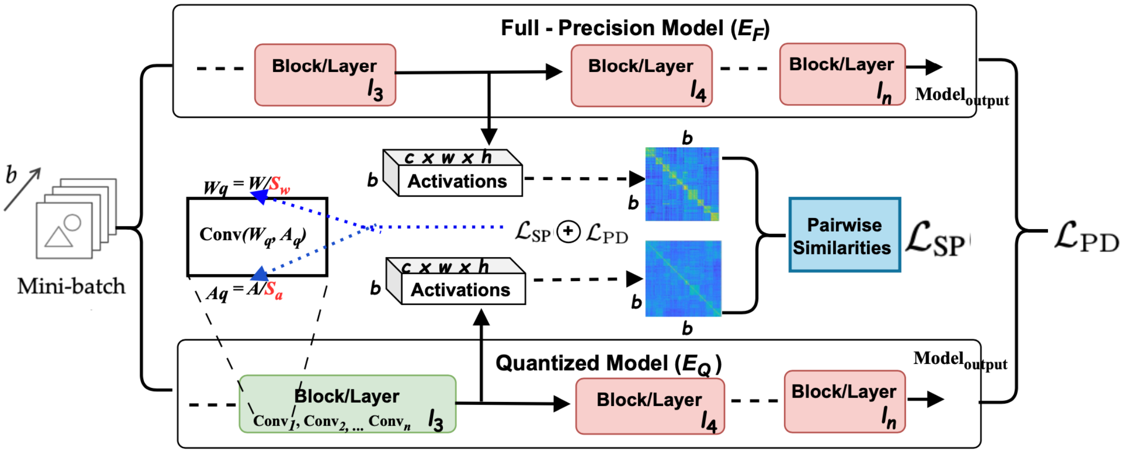


Figure 1. Overview of the proposed SPQ: A pipeline of the proposed method to find best activation scaling factors (S_a) to quantize the activation. Green and red rectangles represent the quantized and full-precision layer/block, respectively. L_{SP} is similarity-preserving quantization reconstruction loss that can be combined with different loss functions.

2.4. Quantization Reconstruction Error

This sacrifice of accuracy is associated with the error that occurs when a neural network model is quantized, typically to lower-precision numerical representations. This quantization error can be controlled by quantization reconstruction error, which acts as a regularizer that reduces generalization error by aligning corresponding components of the quantized and full-precision models. AdaRound [18] analyzes that it is not advisable to round full precision weight to its nearest fixed-point value and proposes a novel rounding mechanism that assigns a continuous variable to each weight value, determining whether it should be rounded up or down rather than employing the traditional nearest rounding method. BRECQ (Block REConstruction Quantization) [13] establishes block-wise reconstruction between the full-precision and quantized network outputs, balancing cross-layer dependency and generalization errors. Additionally, it incorporates trainable clipping for activations. Similar methods have been explored in earlier works [1, 8]. AQuant [29] enhances activation quantization strategy and overall quantization performance, although at the cost of increased inference overhead. PD-Quant [16] addresses the discrepancy between the distribution of calibration activations and their corresponding real activations by proposing a technique for adjusting the calibration activations accordingly. Previous research has investigated the influence of the calibration dataset on the performance of quantized models [8]. Besides, Bannee et al. [2] have explored the reconstruction of features by calculating the feature output distance between quantized and full-precision models, making the quantized model mimic the full-precision model. In contrast to the previous methods, in SPQ, the quantized model is not required to mimic the full-precision model in representation space but rather to preserve the pairwise similarities in its own representation space.

3. Methodology

The core concept of the proposed method is exploring important information in the activation map of the full-precision model and transferring this vital information into the quantized model. Moreover, we set $Z = 0$ to eliminate the zero-point offset in Eqs. 2 and 4; this method simplifies the accumulation operation and reduces computational overhead. However, this simplification comes at the cost of a restricted mapping between the integer and floating-point domains, while suitable for one-tailed distributions like Rectified Linear Unit (ReLU) activations, as claimed by Nagel et al. [19].

The proposed method uses a different method to reconstruct each quantized model by layer or block named Block Similarity-Preserving for Post-Training Quantization (SPQ). This method is inspired from [13, 27]. Tung et al. [27] applies similarity preservation at the batch level, focusing primarily on the similarity of the last convolution layers, while SPQ focuses on intermediate blocks. On the other hand, Li et al. [13] focused on applying Fisher information, SPQ focuses on similarity-preservation instead.

3.1. Overview of SPQ

Figure 1 shows the overall procedure of the proposed method. SPQ is a technique for reconstructing a quantized neural network model using knowledge from a full-precision model. Unlike BRECQ [13] and QDrop [31], reconstruction methods match output values or class probabilities, eliminating the difference in the activation output. SPQ aims to preserve the similarity in the relationships between activations in the quantized and full-precision models, such that input pairs' samples and their corresponding relations across the feature spaces that produce similar or dissimilar activations in the quantized and the full-precision models are maintained. In the context of quantization reconstruction error, we hypothesize that aligning the activation patterns produced by the quantized model with those generated by the full-precision model for highly similar input pairs, as measured by a chosen similarity metric, can yield benefits for quantization reconstruction error. Moreover, we focus on integrating similar information into constructing a new data representation in a quantized model, significantly improving quantization reconstruction error tasks. Unlike prior approach such as BRECQ that primarily focuses on quantizing only the backbone network and QDrop that only considers the Feature Pyramid Networks (FPN) [14], the proposed method extends quantization to more additional components within the detection model; Region Proposal Network (RPN) [21], Region of Interest (RoI) [21], and FPN. This results in a fully quantized detector, avoiding potential hardware incompatibility that could arise due to mixed precision. More importantly, the proposed idea can be readily applied to other post-training quantization methods such as BRECQ and PD-Quant [16]. Furthermore, we aim to preserve the original data learning similarity information from the full-precision model. To this end, we use the widely used spatial and channel similarity activation map.

The following gives the formal explanation of SPQ. For a given mini-batch of input pair samples, let the activation map generated by the full-precision model F (resp. the quantized model Q) at a specific layer or block l be $E_F^{(l)}, E_Q^{(l)} \in \mathbb{R}^{(b \times c \times h \times w)}$, where b represents the batch size, c signifies the number of output channels, and h and w denote the spatial dimensions. We define the quantization reconstruction loss that penalizes differences in the L$_2$-normalized inner products of $E_F^{(l)}$ and $E_Q^{(l)}$. Suppose X is either F or Q, the similarity matrix $Z_X^{(l)}$ is calculated as:

$$Z_X^{(l)} \quad = \quad \frac{r\left(E_X^{(l)}\right) \cdot r\left(E_X^{(l)}\right)^\top}{\varGamma}, \tag{5}$$

where $r(\cdot)$ is a reshape function of $E_X^{(l)}$ (details will be explained in subsequent sections), and $\varGamma$ is the normalization factor. Intuitively, entry (i, j) in $Z_X^{(l)}$ encodes the similarity of the activations at l elicited by the i-th and j-th images in the mini-batch. We define the similarity-preserving quantization reconstruction loss as:

$$\mathcal{L}_{\text{SP}}(F, Q) = \frac{1}{b^2} \sum_{(l,l') \in K} \left\| Z_F^{(l)} - Z_Q^{(l')\top} \right\|_f^2, \tag{6}$$

where K collects the corresponding layer pairs (e.g., layer pair at the end of the same block) and $\|\cdot\|_f$ is the Frobenius norm. Eq. 6 represents a summation across all paired layers $(l, l') \in K$, of the mean squared element-wise difference between the Gramian matrices $Z_F^{(l)}$ and $Z_Q^{(l')}$of the full-precision and quantized models, respectively. Finally, the total loss for searching the optimal activation and weight scaling factors for model quantization is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{KD}}(x, \gamma(y)) + \beta \mathcal{L}_{\text{SP}}(F, Q), \tag{7}$$

where β represents the regularization loss imposed on quantization reconstruction loss, and L_{KD} is any Knowledge Distillation loss to regularize the output probabilities of the quantized model.

3.2. Similarity-Preservation Strategy

SPQ reconstructs quantized features through pairwise similarity across spatial, channel, and batch dimensions. This multi-level similarity leverages fine-grained information for effective reconstruction. Further details on this method are discussed in this section.

Batch Similarity-Preservation: Semantically similar images exhibit high pairwise similarity of activation maps, while dissimilar images exhibit low pairwise similarity. This property can be exploited during calibration by measuring pairwise similarities within the activation map of a batch of images obtained from the full-precision model. These relationship similarities among image batches can then be used to guide the calibration of the quantized model. Batch similarity-preservition is computed by a re-shaped function $r_{\texttt{batch}} : \mathbb{R}^{(b \times c \times h \times w)} \rightarrow \mathbb{R}^{(b \times (c \times h \times w))}$. Therefore, $Z_F^{(l)}, Z_Q^{(l)} \in \mathbb{R}^{(b \times b)}$.

Spatial Similarity-Preservation: Unlike batch similarity, spatial pairwise similarity measures the proximity between individual pixels within an image based on pixel-wise correlation. It computes at the image-level by $r_{\texttt{spatial}} : \mathbb{R}^{(b \times (c \times h \times w))} \rightarrow \mathbb{R}^{(b \times (h \times w) \times c)}$. Therefore, $Z_F^{(l)}, Z_Q^{(l)} \in \mathbb{R}^{(b \times (h \times w) \times (h \times w))}$.

Channel Similarity-Preservation: It reshapes features and calculates similarity across channels, resulting in a different output size. A 1×1 convolution ensures compatible channel dimensions before reshaping by $r_{\texttt{channel}} : \mathbb{R}^{(b \times (c \times h \times w))} \rightarrow \mathbb{R}^{(b \times c \times (h \times w))}$. Therefore, $Z_F^{(l)}, Z_Q^{(l)} \in \mathbb{R}^{(b \times c \times c)}$.

Spatial and Channel Similarity-Preservation: It is achieved by fusing spatial and channel pairwise similarities. We compute the quantization reconstruction loss (L_{SP}) by Eq. 7 using transformed activation maps from both types of similarities and linearly combine their individual losses.

3.3. Computational Efficiency

We use post-quantization Bit OPerations (BOPs) to evaluate accuracy-power trade-offs at different bit-width. Unlike prior works [28], BOPs do not guide our quantization, but we measure its efficiency by Eq. 9, following [30].

$$\begin{aligned} \text{MAC} &= c_i \cdot b \cdot h_o \cdot w_o \cdot k_h \cdot k_w \cdot c_o, & (8) \\ \text{BOPs} &= w_b \cdot a_b \cdot \text{MAC}, & (9) \end{aligned}$$

where w_b and a_b are the bit-width of weights and activations. For the MAC (Multiply and ACcumulate) operation in Eq. 8, c_i and c_o are the input and output channel size, h_o and w_o are the output height and width, k_h and k_w are the kernel height and width, and b is the batch size, respectively.

4. Experimental Evaluation

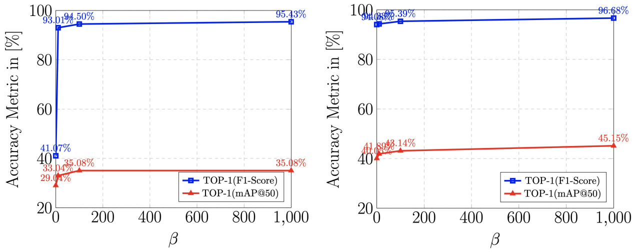
This section presents a comprehensive evaluation of the performance of the proposed algorithm. We begin by outlining the experimental setup and implementation details. Subsequently, we compare our quantization method, evaluated across various low-bit-width configurations, against the state-of-the-art QDrop [31] and BRECQ [13] in our proposed dataset of crop diseases. Finally, we conduct systematic ablation studies to gain deeper insights into the key properties and contributions of our method.

4.1. Implementation Setup

Here, we analyze the inference time for an image with a dimension of $3 \times 4,032 \times 3,024$ on NVIDIA RTX A6000 GPU and Intel Core i9 CPU (2.3GHz 8-Core) to demonstrate that models quantized by the proposed method can reduce the model memory and accelerate inference with negligible accuracy drop. We tested this method on the Faster R-CNN [21] detector under MobileNetV2 [23] and ResNet-50 [7] backbones. The experiments were implemented using the PyTorch [20] framework. After quantizing the model, it was reconstructed block by block to recover the accuracy. The reconstruction was based on finding the optimal scalar factor for the weight and activation feature map. We did not train the quantized weight following the PTQ concepts. The weight rounding scheme adopted in our work adhered to Nagel et al.'s method [18]. For other hyperparameters related to the reconstruction process, such as the number of iterations and loss ratios, we maintained consistency with those reported in QDrop [31] and BRECQ [13]. Notably, we deviated by employing an 8-bit representation for the output of the first layers and detector head in all experiments, which positively impacted accuracy. Batch sizes 16, Adam optimizer [11], and the initial learning rate 0.003 were used.

4.2. Settings

Dataset: We selected maize leaves from three disease classes; namely, *Northern corn Leaf Blight* (NLB), *Fall ArmyWorm* (FAW) and *Maize Streak Virus* (MSV). The dataset comprises four different datasets; more than 18,222 images annotated with 105,735 NLB lesions were collected in the USA [32], and images collected across three Sub-Saharan African countries (Ghana, Uganda, and Namibia) in the field with two different classes: FAW and MSV. The dataset was split into three sets: train, calibration, and validation in the proportion of $70 : 20 : 10$. The train and validation sets were used to train and validate the full precision model. The calibration and validate sets were applied to recalibrate and validate the quantized model.



(a) MobileNetV2 [W8A8]

(b) ResNet-50 [W8A8]

Figure 2. Sensitivity of hyper-parameter β on Faster-RCNN [21] with MobileNetV2 [23] and ResNet-50 [7] backbones, by quantizing weight and activation in 8-bit (W8A8)

Metrics: We measured the network efficiency in four dimensions; *Inference*, *Memory*, *Bit OPerations (BOPs)*, and *Accuracy*. Inference is the time needed to make a prediction, and a smaller value indicates that the model runs faster. Memory measures the memory footprint required to run the model, and a smaller value indicates smaller memory size consumption. We measured BOPs for a single forward pass of the model using Eq. 9 to quantify the computational efficiency. Regarding Accuracy, F1-Score and mean Average Precision (mAP) were used. F1-Score is the harmonic mean of Precision and Recall for the optimized confidence score threshold. On the other hand, mAP provides a comprehensive view of the trade-off between Precision and Recall. We specifically applied mAP@50, considering the nature of some crop diseases. These diseases, with their distinct and well-defined symptoms, are relatively more straightforward to detect. However, certain diseases pose a challenge. Their symptoms are more diffused and vary in appearance, making their detection complex.

4.3. Effects of Hyper-Parameters

Note that β is a weight parameter for balancing the regularization loss imposed on quantization reconstruction loss on similarity-preserving loss in Eq. 7. The overall results are shown in Fig. 2. It shows that the increase of β can improve the disease detection results, reflecting the quantized model's good generalization ability. We can also see the quantized model achieves the best performance when β = 1,000. In other words, the larger β is, the more the quan-

Table 1. Comparison of Faster-RCNN [21] between Full-Precision (FP-32) and quantized models across various bit-width from W8A8 to W2A2, based on MobileNetV2 [23] and ResNet-50 [7] backbones on the calibration dataset. Note that the Inference and Throughput were measured using an image with shape $[3 \times 4,032 \times 3,024]$ on the CPU.

	MobileNetV2					ResNet-50				
	Full-Precision	Quantized model				Full-Precision	Quantized model			
	FP-32	W8A8	W6A6	W4A4	W2A2	FP-32	W8A8	W6A6	W4A4	W2A2
Inference [ms]	1,373.52	1,029.34	1,023.23	1,017.03	1,011.02	2,219.13	1,962.21	1,609.00	1,322.01	1,019.10
Throughput [#images/sec]	0.73	0.97	0.98	0.98	0.99	0.45	0.51	0.62	0.76	0.98
Model Memory [MiB]	346.94	111.26	91.63	72.04	52.39	164.18	45.67	35.80	25.92	16.05
Bit OPerations (BOPs) [T]	26.80	1.67	0.94	0.42	0.10	90.71	5.67	3.19	1.42	0.35
Top-1 mAP@50 [%]	37.10	37.11	35.08	6.67	0.04	46.72	45.15	45.00	43.27	12.01
Top-1 F1-Score [%]	97.46	97.47	95.43	41.50	16.34	97.69	96.68	97.13	96.53	25.01

tized model would learn from the similar activation in the full-precision model. That is to say, the greater β is, the greater the effect of similarity preservation loss is. We set β to 0, 10, 100, and 1,000. From Fig. 2a, we found that the similarity-preservation improved 0.93% on Top-1 (F1-Score). When β = 1,000, the quantized model did not improve on Top-1 (mAP@50). When β gradually decreased from 10 to 0 ($\beta = 0$, without similarity-preservation), the improvement of mAP@50 and F1-Score became smaller and smaller, from 29.04% to 33.04% and 41.07% to 93.01% respectively for Top-1 (mAP@50) and Top-1 (F1-Score). Figure 2b further supports this observation. The similarity-preservation in the quantized model improved by 2.01% Top-1 (mAP@50) and 1.29% Top-1 (F1-Score) when β increased from 100 to 1,000. As β decreased from 100 to 10 and 0 (without similarity-preserving), the improvements gradually decreased to 43.14%, 41.89%, and 40.05% for mAP@50 and 95.39%, 94.33%, and 94.08% for F1-Score, respectively. This proves that introducing similarity-preservation can learn the valuable feature information between full-precision and quantized models to improve the quantization reconstruction error results.

4.4. Evaluation Results

Table 1 shows the results of the detection model with different backbones and quantizer bit-width. Before being quantized, the model achieved 37.10% for Top-1 (mAP@50) and 97.46% for Top-1 (F1-Score) on MobileNetV2 [23], and 37.10% for Top-1 (mAP@50) and 97.46% for Top-1 (F1-Score) on ResNet-50 [7] backbones. The results show that full precision (FP-32) requires both models' significant inference time and memory footprint.

MobileNetV2: When the quantizer bit-width was set to W8A8 (Weight bit = 8 and Activation bit = 8), the quantized MobileNetV2 improved the full-precision by 0.01% for both Top-1 (mAP@50) and Top-1 (F1-Score). Moreover, we continued to see significant improvement in model efficiency, with a good balance between the F1-Score and the detector. We reduced the inference and throughput by 1.33× faster and BOPs to 1.67. The quantized MobileNetV2 memory footprint improved by 3.78× smaller, and the BOP improved by more than 16.05× compared to FP-32, which is an outstanding achievement even though increasing the accuracy in the MobileNetV2. The results show that the models effectively balanced accuracy, speed, and memory footprint in W8A8 bit-width quantizer. When the quantizer bit-width reached W6A6, the quantized MobileNetV2 efficiency significantly increased and was lightweight, which can be confirmed by the BOP reaching a lower value of 0.94, making the model 28.51× efficient. However, the accuracy significantly dropped, achieving 35.08% Top-1 (mAP@50) and 95.43% for Top-1 (F1-Score). Nevertheless, the results show that the models achieved an effective improvement of speed and memory footprint, leading the model on both quantizer bit-width suitable for disease detection despite a decrease of accuracy by 2.02%, Top-1 (mAP@50) and 2.03% Top-1 (F1-Score). When the bit-width reached W4A4, the model's efficiency was greatly optimized and lightweight. However, the accuracy significantly dropped, achieving 6.67%, Top-1 (mAP@50) and 41.50% Top1 (F1-Score) making the model lose balance between accuracy and efficiency. Similarly, when the bit-width W2A2 was applied, the model was almost wholly damaged, reaching 0.04%, Top-1 (mAP@50) and 16.34% Top1 (F1-Score), leading the model on both bit-width unsuitable for disease detection despite significant efficiency improvement.

ResNet-50: The quantized ResNet-50 [7] supports this observation. At a bit-width of W8A8, the quantized model achieved a slight reduction accuracy by 1.57% in Top-1 (mAP@50) and 1.01% Top-1 (F1-Score) compared to the full-precision. Notably, it offered significant efficiency gains, with a 1.13× faster inference speed, 16.00× reduction in BOP, and a 3.59× smaller memory footprint. These improvements successfully balance accuracy, speed, and memory footprint, even at relatively low bit-widths. However, further reducing the bit-width to W6A6 significantly improved efficiency 28.44× reduction in BOPs, making

Table 2. Benchmark of Faster-RCNN [21] from bit-width W8A8 to W2A2, based on MobileNetV2 [23] and ResNet-50 [7] backbone on the calibration dataset. The best score in each column is bold-faced.

(a) Top-1 (mAP@50)

	MobileNetV2					ResNet-50				
	FP-32	W8A8	W6A6	W4A4	W2A2	FP-32	W8A8	W6A6	W4A4	W2A2
QDrop [31]	37.10	37.10	34.79	5.39	0.02	46.72	45.10	44.56	42.17	11.57
BRECQ [13]	37.10	36.08	33.11	4.11	0.00	46.72	44.03	42.54	41.47	10.49
SPQ (Proposed)	37.10	**37.11**	**35.08**	**6.67**	**0.04**	46.72	**45.15**	**45.00**	**43.27**	**12.01**

(b) Top-1 (F1-Score)

	MobileNetV2					ResNet-50				
	FP-32	W8A8	W6A6	W4A4	W2A2	FP-32	W8A8	W6A6	W4A4	W2A2
QDrop [31]	97.46	**97.47**	95.29	**41.52**	16.00	97.69	**96.67**	**97.13**	95.41	23.26
BRECQ [13]	97.46	96.44	94.11	39.98	15.31	97.69	94.67	95.11	94.81	22.22
SPQ (Proposed)	97.46	**97.47**	**95.43**	41.50	**16.34**	97.69	96.66	**97.13**	**96.53**	**25.01**

the model lighter and faster. This resulted in reduced accuracy to 45.00% Top-1 (mAP@50) and 97.13% Top-1 (F1-score), indicating a trade-off between accuracy and efficiency. Nonetheless, the substantial improvements in speed and memory footprint make the quantized model with W6A6 bit-width a viable option for disease detection applications for this backbone, despite a slight decrease in accuracy of 1.72% Top-1 (mAP@50) and 0.56% Top-1 (F1-score). At W4A4, while efficiency significantly improved, accuracy decreased to 43.27%,Top-1 (mAP@50) and 96.61% Top1 (F1-Score), compromising the accuracy-efficiency balance. The W2A2 further decreased accuracy of 8.54%, Top-1 (mAP@50) and 26.74% Top1 (F1-Score), rendering it unsuitable for disease detection despite substantial efficiency gains.

4.5. Performance Analysis

We extensively compared SPQ with various PTQ algorithms across various bit-width configurations without misleading QAT comparisons due to their inherent training differences. Notably, SPQ consistently outperformed other methods, particularly at low bit-widths. Encouraging only the quantization to mimic different aspects of the full precision representation space to optimize activation scaling factors proved insufficient for low bit-widths scenarios. Therefore, SPQ focuses on integrating similar information into constructing a new data representation in a quantized model, significantly improving the optimization of rounding values and activation scaling factors. We benchmark against QDrop [31] and BRECQ [13], the strongest-performing PTQ methods across various bit-width from W8A8 to W2A2. To ensure consistency, we applied Spatial and Channel Similarity-Preservation as described in Section 3. All the methods used MobileNetV2 [23] and ResNet-50 [7] backbones. Tabs. 2a and 2b summarize the results respectively for Top-1 (mAP@50) and Top-1 (F1-Score). The results demonstrate substantial improvements achieved by SPQ compared to strong PTQ baselines. The proposed SPQ algorithm has demonstrated substantial improvements compared to strong PTQ baselines across various performance metrics such as Top-1 (mAP@50) and Top-1 (F1-Score). This comprehensive study, conducted with meticulous attention to bit-width detail, provides a confident assessment of the proposed algorithm's performance under different model architectures.

MobileNetV2: The gains were modest when the bit-widths were set to W8A8 under MobileNetV2 [23]. However, they became more pronounced at lower bit-widths. For instance, SPQ outperformed QDrop and BRECQ at W8A8 settings quantization, by improving MobileNetV2 [23] by 0.01% and 1.03% for Top-1(mAP@50), respectively. Furthermore, when we set the bit-widths to W6A6, we observed significant improvements of Top-1 (mAP@50) (0.29%, 1.97%) and Top-1 (F1-Score) (0.14%, 1.32%) compared to QDrop and BRECQ, respectively. This trend continues at even lower bit-widths W4A4 where SPQ achieved gains and negligible decrease of Top-1(mAP@50) (1.28%, 2.56%) and Top-1 (mAP@50) (−0.02%, 1.52%) for QDrop and BRECQ, respectively. On the other hand, we noticed that MobileNetV2 is sensitive to lower bit-widths. Despite this inherent limitation, SPQ still demonstrate consistent superiority in W2A2 bit-widths over QDrop and BRECQ by Top-1(mAP@50) (0.02%; 2.56%) and Top-1(F1-Score) (−0.02%, 1.52%), respectively.

ResNet-50: Similar to MobileNetV2, ResNet-50 [7] exhibits modest accuracy improvements with SPQ at higher bit widths W8A8. Specifically, we observed gains of Top-1(mAP@50) (0.05%, 0.06%) but a negligible decrease of Top-1(F1-Score) (-0.01%, 0.10%) compared to QDrop and BRECQ, respectively. This suggests that existing quantization techniques can effectively capture a significant portion of the quantifiable benefits at higher precisions. However, the advantage of SPQ becomes more pronounced as bit-widths are reduced. Furthermore, when we set bit-widths to W6A6, we achieved statistically significant improvements of Top-1 (mAP@50) (0.04%, 0.6%) compared to QDrop and BRECQ, respectively, while for Top-1 (F1-Score) only improved 1.03% compared to BRECQ. This trend continues at even lower precisions at W4A4, where SPQ surpasses QDrop and BRECQ by a wider margin for Top-1 (mAP@50) (1.1%, 1.8%) and Top-1 (F1-Score) (1.12%, 1.72%), respectively. Unlike MobileNetV2, ResNet-50 demonstrates some resilience to aggressive quantization at W2A2 bit-widths. SPQ still achieves consistent improvements over QDrop and BRECQ at this shallow precision setting, with gains of Top-1 (mAP@50) (0.44%, 1.52%) and Top-1 (F1-Score) (1.75%, 2.79%), respectively.

The experiment underlines the significance of SPQ's optimization strategy. Moreover, SPQ requires no additional computation for inference after optimization, ensuring efficiency. Additionally, SPQ prioritizes hardware compatibility through uniform bit-width quantization, this approach may only consistently achieve optimal accuracy or efficiency for some tasks. To address this, we leverage hyper-parameterization for bit-width selection. Furthermore, SPQ focuses more on sensitive and complex tasks such as object detection, where the proposed method was intensively experimented. To the best of our knowledge, this work is the first to achieve PTQ of an entire object detection model at low bit-widths W6A6 and W4A4 to a usable level, as demonstrated in Tab. 2b. Unlike BRECQ and QDrop, SPQ extends quantization to more additional components within the detection model: Region Proposal Network (RPN) [21], Region of Interest (RoI) [21] and Feature Pyramid Networks (FPN) [14]. Which results in a fully quantized detector, eliminating potential hardware compatibility issues that could arise from mixed precision (quantized and unquantized weights) within each component of the model. Consequently, the entire quantized model becomes more hardware-agnostic and easier to deploy on various platforms.

4.6. Effect of Loss Functions

We conducted experiments of quantization reconstruction error based only on similarity-preserving loss in Eq. (7) using ResNet-50 [7]. We conducted extensive experiments in different bit-widths, from W8A8 to W6A6 bits quantization, for all layers/blocks except for the first layers and detector head. Furthermore, we performed an extensive analysis based on two primary metrics, Top-1 (mAP@50) and Top-1 (F1-Score). From Tab. 3, we can observe that Similarity-Preserving loss (L_{SP}) has stable accuracy improvement at all bits, which implies that the generalization of quantization error in L_{SP} consistently outperformed the L_{KD} in all settings.

Table 3. Ablation study of loss functions in quantized models across various bit-width from W8A8 to W6A6, based on ResNet-50 [7] backbone on the calibration dataset. The best score in each column is bold-faced.

	ResNet-50					
	Top-1 mAP@50 in [%]			Top-1 F1-Score in [%]		
	FP-32	W8A8	W6A6	FP-32	W8A8	W6A6
$\mathcal{L}_{\text{KD}}$	46.72	40.05	38.89	97.69	94.08	94.13
$\mathcal{L}_{\text{SP}}$	46.72	42.10	42.09	97.69	95.67	95.31
$\mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{SP}}$	46.72	**45.15**	**45.00**	97.69	**96.66**	**97.13**

5. Conclusion

This paper revealed that encouraging only the quantized model to mimic different aspects of the full-precision model to optimize activation scaling factors is insufficient for low-bit scenarios and complex datasets. Meanwhile, we observed that capturing global structure in activation map information and preserving the original pairwise similarities between the activation map points in the full-precision model and the quantized model in the embedding space is promising. The proposed method is particularly suitable for problems sensitive to sample similarity, such as classification, detection, and drug similarity in recommender systems, and disease detection in healthcare informatics. The proposed method can improve the performance quantization significantly. This is because it is based on similarity, while other methods are based on Euclidean distance, which is unsuitable for complex tasks such as detection. Furthermore, our hardware-friendly method allows bit homogeneity through bit-width hyper-parameters.

Future work could explore how to improve our method, such as combining with binary quantization. We will also consider how to apply our method to model pruning.

Acknowledgement. This work was partly supported by JSPS KAKENHI JP21H0355.

References

[1] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *Advances Neural Information Processing Systems*, 32:7948–7956, 2019. 3

[2] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. ACIQ: Analytical Clipping for Integer Quantization of

neural networks. *Computing Research Repository arXiv Preprints*, arXiv:1810.05723, 2018. 3

[3] Emily L. Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *dvances Neural Information Processing Systems*, 27:1269–1277, 2014. 1

[4] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. *Computing Research Repository arXiv Preprints*, arXiv:1902.08153, 2019. 2

[5] Food and Agriculture Organization of the United Nations. New standards to curb the global spread of plant pests and diseases, 2021. https://www.fao.org/news/story/en/item/1187738/icode/ (Visited: 04-July-2023). 1

[6] Food and Agriculture Organization of the United Nations. Agricultural production statistics. 2000–2020. FAOSTAT Analytical Brief Series, No. 41, 2022. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceeding of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5, 6, 7, 8

[8] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training quantization with small calibration sets. In *Proceedings of the 38th International Conference on Machine Learning*, 4466–4475, 2021. 3

[9] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2704–2713. 1, 2

[10] Kamaljit Kaur and Manpreet Kaur. Prediction of plant disease from weather forecasting using data mining. *International Journal on Future Revolution in Computer Science Communication Engineering*, 4(4):685–688, 2018. 2

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Computing Research Repository arXiv Preprints*, arXiv:1412.6980, 2017. 5

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012. 2

[13] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *Proceedings of the 9th International Conference on Learning Representations*, 2021. 1, 2, 3, 5, 7

[14] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *Computing Research Repository arXiv Preprints*, arXiv:1612.03144, 2016. 4, 8

[15] Bin Liu, Yun Zhang, DongJian He, and Yuxiang Li. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 10(1):1–11, 2018. 2

[16] Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. PD-Quant: Post-Training Quantization Based on Prediction Difference Metric. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24427–24437, 2023. 3, 4

[17] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*, pages 1325–1334, 2019. 2

[18] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? Adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7197–7206, 2020. 1, 3, 5

[19] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *Computing Research Repository arXiv Preprints*, arXiv:2106.08295, 2021. 2, 3

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8024–8035, 2019. 5

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99, 2015. 4, 5, 6, 7, 8

[22] Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif. Plant disease detection and classification by deep learning. *Plants*, 8(11):468–479, 2019. 1

[23] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *Computing Research Repository arXiv Preprints*, arXiv:1801.04381, 2018. 5, 6, 7

[24] Yuzhang Shang, Dan Xu, Bin Duan, Ziliang Zong, Liqiang Nie, and Yan Yan. Lipschitz continuity retained binary neural network. In *Proceedings of the 18th IEEE/CVF International Conference on Computer Vision*, pages 10655–10664, 2022. 2

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computing Research Repository arXiv Preprints*, arXiv:1409.1556, 2016. 2

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2

[27] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019. 3

[28] Mart van Baalen, Brian Kahne, Eric Mahurin, Andrey Kuzmin, Andrii Skliar, Markus Nagel, and Tijmen Blankevoort. Simulated quantization, real power savings. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2756–2760, 2022. 5

[29] Changbao Wang, DanDan Zheng, Yuanliu Liu, and Liang Li. Leveraging inter-layer dependency for post -training quantization. In *Advances Neural Information Processing Systems*, 2022. 3

[30] Ying Wang, Yadong Lu, and Tijmen Blankevoort. Differentiable joint pruning and quantization for hardware efficiency. *Computing Research Repository arXiv Preprints*, arXiv:2007.10463, 2020. 5

[31] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. QDrop: Randomly dropping quantization for extremely low-bit post-training quantization. *Computing Research Repository arXiv Preprints*, arXiv:2203.05740, 2022. 3, 5, 7

[32] Tyr Wiesner-Hanks and Mohammed Brahimi. Image set for deep learning: Field images of maize annotated with disease symptoms, 2019. https://osf.io/p67rz/ (Visited: 04-July-2023). 5

[33] Xin Xia, Xuefeng Xiao, Xing Wang, and Min Zheng. Progressive automatic design of search space for one-shot neural architecture search. In *Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision*, page 2455–2464, 2022. 2

[34] Qian Yan, Qian Yan, Qian Yan, Bing Wang, Bing Wang, and Jun Zhang. Apple leaf diseases recognition based on an improved convolutional neural network. *Sensors*, 20(12):3535–3547, 2020. 2

[35] Xihai Zhang, Yue Qiao, Fanfeng Meng, Chengguo Fann, and Mingming Zhang. Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access*, 6:30370–30377, 2018. 1