Abstract—Quantizing the floating-point weights and activations of deep convolutional neural networks to fixed-point representation yields reduced memory footprints and inference time. Recently, efforts have been afoot towards zero-shot quantization that does not require original unlabelled training samples of a given task. These best-published works heavily rely on the learned batch normalization (BN) parameters to infer the range of the activations for quantization. In particular, these methods are built upon either empirical estimation framework or the data distillation approach, for computing the range of the activations. However, the performance of such schemes severely degrades when presented with a network that does not accommodate BN layers. In this line of thought, we propose a generalized zero-shot quantization (GZSQ) framework that neither requires original data nor relies on BN layer statistics. We have utilized the data distillation approach and leveraged only the pre-trained weights of the model to estimate enriched data for range calibration of the activations. To the best of our knowledge, this is the first work that utilizes the distribution of the pre-trained weights to assist the process of zero-shot quantization. The proposed scheme has significantly outperformed the existing zero-shot works, e.g., an improvement of ~33% in classification accuracy for MobileNetV2 and several other models that are w/ & w/o BN layers, for a variety of tasks. We have also demonstrated the efficacy of the proposed work across multiple open-source quantization frameworks. Importantly, our work is the first attempt towards the post-training zero-shot quantization of futuristic unnormalized deep neural networks.

Index Terms—Data distillation, deep convolutional neural networks (CNNs), model compression, post-training quantization

I. INTRODUCTION

Deep CNNs are renowned for their tremendous capabilities to learn robust and complex features to produce remarkable results. Over the last decade, they have been widely adopted in various fields such as Computer Vision [1], Speech Processing [2], Natural Language Processing [3], etc., spanning from system to on-device platforms. However, several modern deep neural networks have an enormous size, which makes it challenging to deploy those on real-time resource constraint devices. For e.g., LeCun et al. [4] proposed LeNet5 model that has approx. 400K weights. About two decades later, AlexNet [5] and VGG-16 [6] were proposed that have more than 60M, 130M weights, respectively. Each of these models, when stored in 32-bit floating-point (FP32) format, requires excessive storage, e.g., VGG16 [6], that has a size of 130M×4Bytes ≃ 520MBytes. Coates et al. [7] even proposed a model that has 11B weights. Such models may also result in higher inference latency, processing capacity, energy consumption, and difficulty in training [8].

Furthermore, a majority of the inference specific accelerators, such as Google Edge TPU, support only the quantized models. For e.g., MobileNetV2 [9], that takes 51 milliseconds (ms) per inference for ImageNet [10] classification on a Desktop CPU, takes only 2.6 ms on a device with Google Edge TPU, about 20× faster. Therefore, it becomes necessary to compress the large-sized models in order to accommodate them on resource constraint devices and accelerators for an efficient real-time computing.

Existing methods for deep model compression are primarily based on matrix factorization, pruning, quantization, and designing deep CNN’s with fewer parameters [9, 11–19]. Also, quite a few schemes [20–22] have been proposed that train a deep CNN based on Bayesian methods in order to aid the quantization and pruning at a later stage. However, in this work, we focus on quantization and propose a novel principled zero-shot approach.

Background: Quantization allows us to store or process the tensors at lower and ultra-low bit precisions, e.g., <4, 8, 16>-bit fixed-point from FP32 format [23]. It yields more resource-efficient integer operations. Given a tensor $x$ (in FP32), its quantized value $Q(x)$ can be approximated as

$$Q(x) = \text{round}(x/\Delta + z),$$  

where $\Delta$, $z$ denote scale and zero-point offset, respectively. The $\Delta$ can be computed as

$$\Delta = n/(\max(x) - \min(x)),$$  

where $n = 2^b - 1$ with $b$ denoting lower bit precision. The zero-point offset then can be computed as follows

$$z = \Delta \times \min(x).$$
Overall, the quantization schemes can be classified into (a) Per-tensor (QS-PT), and (b) Per-channel (QS-PC), alias Per-axis. Per-tensor quantization requires $\Delta$ and $z$ to be computed across the tensor. Whereas in per-channel, they are calculated across each channel of the tensor. Further, they are categorized as symmetric if $z$ is 0, asymmetric (alias affine) otherwise [23]. We refer the readers to [23, 24] for a detailed review of various quantization schemes.

In the case of post-training quantization, $\Delta$ and $z$ are computed for both weights and activations prior to inference. Their computations (see Eqns. 2, and 3) depend on range of the input tensor $x$, i.e., $\min(x)$, and $\max(x)$. We refer the process of estimating the range of the input tensor as “range calibration” in the rest of the paper. Given a pre-trained model in FP32, range calibration of the weights can be achieved without any difficulty. However, for activations\(^1\), one requires access to the original unlabelled training samples. Once the dataset, either in full or limited, is available, the activations can be generated, followed by computing the $\Delta$ and $z$ of the same. Based on the availability of the original unlabelled training samples, either in full, limited, or none, the recent works can be categorized into (a) quantization-aware training [23–30], (b) requires limited data [15, 31–33], and (c) data-free approaches [34–38], respectively.

**Challenges and Motivation:** Recently, much attention has been given to the data-free approaches [34–36, 38] since the availability of the original data is infeasible in some cases. For e.g., Medical Imaging, where the user’s privacy is prioritized above all. These schemes may also be analogously referred to as zero-shot approaches\(^2\). For estimating the $\Delta$ and $z$ of the activations w/o utilizing the original data, these zero-shot schemes mainly rely on the learned BN [39] layer’s parameters. The long-term statistics of BN [39] layer may consist of some information about the original training samples [36], which may be useful for range calibration of the activations.

BN [39] layer in a deep CNN is an indispensable component that is believed to improve the generalization, accelerate convergence, aid higher learning rate, and stabilize the training process [40]. However, in recent studies [40–42], it has been shown that none of these benefits is unique to the BN [39] layer. Also, BN [39] parameters are folded into the weights and biases of convolution and fully-connected layers for an efficient inference, which is a widely adopted practice [23]. The BN [39] folding introduces high variance into the weights of the pre-trained model, that makes quantization difficult [23]. Hence, the existing zero-shot approaches [34, 35] fail when the BN [39] layer is either folded or absent in the network.

Now, an obvious question that arises, is, “what else can be utilized from the pre-trained model that may represent original training samples, given the zero-shot condition holds.” Perhaps, the pre-trained weights. For any given task, weights are learned to extract the imperative features from the input training samples in order to achieve the desired objective with minimal penalty [43].

Further, when the BN [39] layer is folded, the introduced variance in the weights may have some auxiliary information about the distribution of the original data. Both the pre-trained weights and folded-variance have not been exploited in any of the recent works [15, 23–35]. Therefore, in this work, we have proposed a generalized zero-shot approach, namely GZSQ, that neither requires original training samples nor depends on BN [39] layer parameters for range calibration of the activations. The proposed GZSQ acts as an API call and is built upon the data distillation framework that only considers the pre-trained weights of the model. In particular, GZSQ approximates the substitutes for BN statistics for each layer by only utilizing the pre-trained weights of the network. Hence, it eliminates the requirement of the BN layers in the model if one wants to incorporate data distillation for the range calibration of the activations under zero-shot condition. We have evaluated the proposed method across multiple quantization frameworks with both QS-PC and QS-PT schemes. Further, in our experiments, we have considered the models that are w and w/o BN [39] layers, including the BN [39] folding in the former case, for the tasks of classification [10, 44] and object detection [45].

The rest of the paper is organized as follows: Section II briefly reviews the related existing works. Section III presents the proposed GZSQ framework. The experimental details, results, and ablation study are presented in Sections IV, V, and VI, respectively. Lastly, Section VII summarizes the work.

**II. RELATED DEVELOPMENTS**

Besides quantization, several works have been proposed to address the restricted memory footprints and inference latency of the modern deep CNNs. Going orthogonally to quantization, these methods are based on — efficient neural architecture design, knowledge distillation, pruning, hardware co-design, and matrix factorization.

**Architecture design-based** methods [9, 16–19] proposed the deep CNNs that have fewer number of parameters using Neural Architecture Search (NAS) [46], for e.g., MobileNetV2 [9]. Knowledge distillation based methods [47–49] extracted the latent information from the original model using student-teacher paradigm and proposed the sparse version of the model. To reduce the number of trainable parameters, pruning [11–14] sets the insignificant weights to zero. This includes methods based on Hessian matrix of the loss functions [50, 51], and iterative pruning [52]. Kwon et al. [53] proposed to co-design the deep CNNs and neural net accelerators [54] to achieve significant performance gain. Matrix factorization based schemes [55, 56] decompose the weight matrix into the low-rank (L) and sparse (S) matrices, where both L and S require less storage. Later, Han et al. [57] proposed a method based on the ensemble of pruning, quantization and Huffman coding [58]. Recently, He et al. [59] proposed the reinforcement learning-based method for model compression. However, here, we focus on existing quantization works based on the requirement and availability of the original training samples.

**Require full training set:** Quantization reduces the memory footprints of the deep CNNs by reducing the bit precisions

---

\(^1\) interchangeably referred to as *features* in the paper

\(^2\) since they do not require original data samples
of weights and activations. However, it may also introduce noise and lead to the significant performance degradation, especially when quantizing FP32 to ultra-low bit precisions. To address this, several methods [23–30, 60, 61] follow the trivial approach for quantization by performing the Quantization-aware Training (QAT) [24]. These approaches may result in the lowest degradation in final accuracy. However, it requires a full training set to train the model, which may not be possible in some cases, e.g., Medicine or Satellite Image-based systems, due to privacy. Besides the availability of original data, QAT-based schemes are time and resource consuming.

**Require limited data:** To overcome this drawback, a few methods [15, 31–33], based on weights factorization and channel splitting, have been proposed, requiring limited training samples. In particular, Choukroun et al. [32] proposed a method, namely OMSE, that minimizes the $\ell_2$ distance between quantized and original tensor. Also, Banner et al. [31] proposed a method called ACIQ that analytically computes the clipping range and per-channel bit allocations for deep CNNs. However, per-channel quantization for activations is infeasible in practice. Zhao et al. [15] proposed a method called OCS to solve the problem of outlier channel.

**Zero-shot approaches:** To overcome the drawback of original data dependency, either in full or limited, Nagel et al. [34] proposed a data-free quantization method, namely DFQ. It is based on the weights equalization and bias correction. However, to estimate the $\Delta$ and $z$ of the activations, DFQ entirely depends on the BN’s [39] shift ($\beta$) and scale ($\gamma$) parameters. The $j$th layer activation ranges are first calculated as $\bar{\beta}_j \pm n \cdot \gamma_j$, where $n$ has been empirically set to 6. Later, $\Delta_j$ and $z_j$ have been computed using Eqsns. 2 and 3, respectively. Subsequently, following [36, 37], Cai et al. [35] proposed a zero-shot framework, called ZEROQ, to generate the data that matches the statistics of the training set. For this, ZEROQ [35] has utilized the BN [39] layer’s statistics, as

$$\min_{y^r} \sum_{i=0}^{L} \| \mu_i^r - \mu_i \|^2 + \| \sigma_i^r - \sigma_i \|^2,$$

where $\mu_i^r/\sigma_i^r$ are the mean/std-dev of the distilled data’s ($y^r$) distribution at layer $i$, and $\mu_i/\sigma_i$ are the corresponding mean/std-dev parameters stored in the $i$th BN [39] layer. The distilled data $y^r$ then has been used to infer the range of the activations to estimate the $\Delta$ and $z$. Later, Xu et al. [38] proposed a generative approach [62] to produce a fake data for range calibration by exploiting the classification boundary knowledge and BN [39] statistics. The proposed generator architecture is inspired from AC-GAN [63], which also results in additional memory overhead.

Recently, a shift in trend has been observed towards the Mixed-Precision Quantization (MPQ) [64–66], where a different bit-precision is set adaptive to each layer in the network. However, it further raises another level of difficulty in finding the optimal bit-width for each layer, given a predetermined support by the hardware accelerators to MPQ. Along with DFQ [34], these best-published works fail when the model does not comprise of BN [39] layers. In that case, it becomes difficult to either utilize the distillation approach or BN [39] parameters ($\beta, \gamma$) and infer activation ranges without original training samples.

Moreover, the trend [40–42] observed in the past two years shows that the community has started to look for alternatives to BN [39]. Therefore, it should be mentioned that in the coming years, a majority of the cutting-edge deep CNNs may not consist of BN [39] layers at all. This work is the first attempt towards the quantization of such futuristic models.

**Our Contributions:** We present a novel solution that allows us to utilize the distillation approach, even if the BN [39] layers are folded or absent in a model. For this, we first estimate the substitutes to BN statistics ($\mu, \sigma$) for every layer by utilizing the pre-trained weights of the model only. We divide this substitutes estimation phase into (a) statistics estimation, followed by (b) empirical statistics adjustment sub-phases. We then perform the data distillation, similar to in Eqn. 4, by utilizing the estimated substitutes with a novel objective function. Essentially, our key contributions are fourfold:

- We propose a generalized zero-shot quantization (GZSQ) framework that neither requires original data nor relies on BN [39] layers statistics of the models. In other words, we leverage the pre-trained weights of the model “only” and estimate an enriched data which can be used for range calibration of the activations.
- We also propose to use the absolute Z-score based loss function over $\ell_1$ and $\ell_2$ norms or Kullback-Leibler (KL) divergence for an efficient data distillation.
- To test the generalization proficiency of the proposed GZSQ, we have evaluated our distillation approach across multiple open-source quantization frameworks. We have also benchmarked the models that are w & w/o BN [39] layers on the tasks of classification and object detection.
- To appraise the efficacy of the proposed method, we have shown the results on a different problem domain, e.g., Medical Imaging. Moreover, we have presented an ablation study to demonstrate the effect of various cost functions and BN [39] parameters folding before & after data distillation.

III. PROPOSED APPROACH

| Symbol | Meaning |
|-------|--------|
| $T$ | A tensor that $\in \mathbb{R}^{w \times h \times c}$ |
| $\mu_T, \sigma_T$ | Channel-wise mean and std-dev of $T$, respectively |
| $\hat{F}$ | A pre-trained FP32 model with $N$ conv2d layers |
| $\hat{W}$ | Pre-trained conv2d weights (in FP32) of $\hat{F}$ |
| $W_n$ | Weights of $n^{th}$ conv2d layer in $\hat{F}$ |
| $F_n^{\beta, \gamma}$ | $n^{th}$ layer activations in $\hat{F}$ |
| $\mu_{\hat{F}_n^{\beta, \gamma}}, \sigma_{\hat{F}_n^{\beta, \gamma}}$ | Mean and std-dev of $F_n^{\beta, \gamma}$ |
| $\mu_{BN_n^{\beta, \gamma}}, \sigma_{BN_n^{\beta, \gamma}}$ | Substitutes of BN [39] statistics |
| $K_n^{\beta, \gamma}$ | Set consisting of BN [39] substitutes |
| $\mathcal{N}(\mu, \sigma)$ | Gaussian with $\mu$ mean and $\sigma$ std-dev |

In this section, following the notations in Table I, we describe the regime of operations of the proposed GZSQ scheme.
A. Statistics Estimation

We first describe the process of estimating $\hat{\alpha}$ for $F$ w/o using original data samples. Following [67], if $W_{1:N}$ are initialized with a Gaussian distribution ($\mathcal{N}(\mu, \sigma)$) [68], owing to the Central Limit Theorem, $W_{1:N}$ will also converge to $\mathcal{N}(\mu, \sigma)$ at the end of training. We also assume the input $J$ for distilled data estimation to follow $\mathcal{N}(0,1)$.

Now, let us first consider the base case where in $F$, BN [39] layers exist up to layer $n$, where $n \in (1, N)$. The activations of layer $n$ then can be written as

$$f^n = g^n(BN^n[W_n * f^{n-1} + b^n]),$$

where $g^n$ denotes $n$th layer activation function. Knowing that the pre-activations of layer $n$ follow $\mathcal{N}(\mu, \sigma)$ due to $BN^n$ [34], and if $g^n$ is some form of the class of clipped linear activations, such as ReLU or ReLU6, then $f^n$ will also follow a clipped $(i,j)$ Normal distribution, where $i < j$, and $j$ can be $\infty$.

Now, let us assume, from $n + 1$st layer onwards, BN [39] layers do not exist in $F$. The activations of $n + 1$st layer can be written as

$$f^{n+1} = g^{n+1}[W_{n+1} * f^n + b^{n+1}].$$

We know that $W_{n+1}$ and $f^n$ follow $\mathcal{N}(\mu, \sigma)$. Following the property of convolution of two Gaussian distributions $\mathcal{N}(\mu_1, \sigma_1)$, $\mathcal{N}(\mu_2, \sigma_2)$ is also a Gaussian $\mathcal{N}(\mu_1 + \mu_2, \sqrt{\sigma_1^2 + \sigma_2^2})$, the $f^{n+1}$ will also follow a clipped $\mathcal{N}(\mu, \sigma)$ when $b_1:N = 0$. However, we will consider the extra biases introduced due to BN [39] folding in Section III-D.

The substitutes $\mu_{a^{n+1}}$ and $\sigma_{a^{n+1}}$ then can be estimated as follows

$$\mu_{a^{n+1}} = \mu_{W_{n+1}} + \mu_{a^n},$$

$$\sigma_{a^{n+1}} = \sqrt{\sigma_{W_{n+1}}^2 + \sigma_{a^n}^2}.$$  

This can go on for the subsequent layers ($n + 2 : N$) in $F$. Backtracking our initial assumption of BN [39] layers up to $n$th layer in $F$ to the 1st layer, the $\mu_{a^2}, \sigma_{a^2}$ can be estimated following the above procedure.

Now, suppose, even the 1st layer in $F$ is not followed by a BN [39] layer. Then its activations can be written as $f^1 = g^1[W_1 * J]$. Holding to our initial assumption of $J$ to follow $\mathcal{N}(0,1)$, the $f^1$ will also follow $\mathcal{N}(\mu, \sigma)$ and $\mu_{a^1}, \sigma_{a^1}$ can be estimated using Eqn. 7.

B. Empirical Statistics Adjustment

The channel-wise addition in Eqn. 7 is true, iff the number of channels $(C)$ of the addends are same. However, with the recent advancement in deep learning, there exists a variety of networks where the condition holds false [9, 70–72]. We classify such pair of layers into the sets of (a) Expansion when $C(W_{n+1}) > C(a^n)$, and (b) Contraction when $C(W_{n+1}) < C(a^n)$. We empirically define the following set of values as {min, mean ± min, mean, max ± mean, max}, to be used for accommodating the required number of channels in the addend with lower $C$.

Intuitively, a pair of filters of a deep CNN layer may have some correlation (strong or weak), resulting in correlated feature maps [73]. Therefore, the best approximation of the required statistics from the current ones w/o utilizing any explicit information may be made by leveraging any value from the set mentioned above.

For e.g., in the case of expansion, repeating the current statistics across the channels works for most of the networks, except for ShuffleNet [71] and InceptionV3 [72]-based models, where mean($\mu_{a^n}; \sigma_{a^n}$)−min($\mu_{a^n}; \sigma_{a^n}$) works.

Similarly, for contraction, where in most cases mean($\mu_{a^n};\sigma_{a^n}$)−min($\mu_{a^n};\sigma_{a^n}$) works, except for SqueezeNet [74] where mean values are set with mean($\mu_{a^n}$) and for MobileNetV2 [9] where std-dev values are set using mean($\sigma_{a^n}$)+min($\sigma_{a^n}$).

C. Data Distillation

With the set $\hat{\alpha}$ as substitutes to BN [39] statistics, we have followed the distillation approach similar to ZEROQ [35]. For this, we input a random instance of $\mathcal{N}(0,1)$, denoted by $y$, to $F$ and generate set of activations for each layer. We then compute the statistics of each activation as $(\mu_{f^n}, \sigma_{f^n}) \forall n \in 1 : N$, and minimize the distributional difference with the corresponding values in $\hat{\alpha}$. However, instead of utilizing $\ell_1$ or $\ell_2$ norm, or KL divergence, we have considered the absolute difference of Z-score test as a loss function $L_Z(u,v)$ for an efficient knowledge distillation as

$$L_Z(u,v) = \|\mu_u - \mu_v\|/\sqrt{(\sigma_u + s)^2 + (\sigma_v + s)^2},$$

where $u, v$ can be any two tensors with different distributions and $s = 1e^{-6}$. The $\ell_1$ and $\ell_2$ norms are known to favour the sparse and non-sparse data, respectively. It has been widely accepted that in a deep CNN, the first few activations may have high non-sparsity whereas later layers may have high sparsity [75]. The shortcomings of utilizing KL divergence in GZSQ have been discussed in Section VI-B.

Therefore, instead of data or sparsity based optimization, we propose a distributional minimization ($L_Z$) and define the final loss as

$$L_D = \sum_{n=1}^{N} L_Z(f^n, a^n) + L_Z(y, \mathcal{N}(0,1)).$$

We finetune $y$ using the following objective function

$$\arg\min_y L_D,$$

and then use the $y$ (distilled data w/o utilizing BN [39] statistics) to infer the ranges of the activations in $F$.

\textsuperscript{1}In practice, weights of many deep CNNs are initialized with Gaussian version of the Kaiming initialization [69].

\textsuperscript{2}The readers are advised to refer this link for more details.

\textsuperscript{3}The pre-trained models we have studied in this paper do not use bias.

\textsuperscript{4}Similar to ZEROQ [35], it has been set to avoid the division by 0 in Eqn. 8 if $\sigma_u$ and $\sigma_v$ are 0.
Algorithm 1: GZSQ algorithm

Input: \( F \), in FP32
Output: \( \hat{y} \)
Data: None

/* Details of the methods SE, ESA and DD are given in Subsections III-A, III-B, and III-C, respectively. */
1 \( J \leftarrow \mathcal{N}(0,1) \)
2 Define set \( \mathcal{A} \) as \( \{ (\mu_a^1, \sigma_a^1), (\mu_a^2, \sigma_a^2), \ldots, (\mu_a^N, \sigma_a^N) \} \)
3 \( \mu_a^1, \sigma_a^1 \leftarrow SE(0,1, W_1) \)
4 while \( n \in 2 : N \) do
5 \( \mu_a^{n-1}, \sigma_a^{n-1} \leftarrow SE(\mu_a^{n-1}, \sigma_a^{n-1}, W_n) \)
6 \( \mu_a^n, \sigma_a^n \leftarrow SE(\mu_a^n, \sigma_a^n, W_n) \)
7 \( y \leftarrow \mathcal{N}(0,1) \)
8 \( \hat{y} \leftarrow DD(y, \mu, F) \)

D. The Curious Case of Batch Normalization Folding

In practice, for a resource-efficient inference, the BN [39] parameters are folded into the weights and biases of the convolutional and fully-connected layers of a deep CNN. For e.g., folded weights and biases of the \( n^{th} \) convolutional layer in \( F \) can be written as

\[
W_{n,fold} = \gamma_n \cdot W_n / \sigma_n^B
\]

\[
b_{n,fold} = \beta_n - \gamma_n \cdot \mu_n^B / \sigma_n^B,
\]

where \( \gamma_n, \beta_n, \mu_n^B, \sigma_n^B \) represent the scale, shift, long-term mean, and long-term std-dev of the \( n^{th} \) BN [39] layer, respectively. It has been observed during our experiments that the introduced bias, when BN [39] layers are folded before distillation, may lead to a slight (severe in some cases; e.g., ResNet18 and ResNet50; see Section VI-C and Table. XIII) degradation in the accuracy of the quantized models. Therefore, the introduced bias term has to be accommodated into the proposed distillation approach (see Sections III-A, III-B), when folding is performed before distillation. With the \( \sigma_{b,fold} = 0 \), the refined mean in Eqn. 7 can be written as

\[
\mu_{a,n} = \mu_{W_{n,fold}} + \mu_{a,n-1} + b_{f,n}^B.
\]

E. Post-Training Quantization

By utilizing the distilled data \( \hat{y} \) to infer the ranges of the activations, the scale \( \Delta \) and zero-point offset \( z \) can be computed for each layer activation. For \( W \), one can compute the ranges and \( \Delta, z \) as usual. In this paper, we have presented the results for both QS-PC and QS-PT schemes. Also, to test the generalization capacity of the proposed GZSQ scheme, we have incorporated both (a) ZERQ’S [35] quantization simulation\(^7\), and (b) Pytorch’s Post-Training Static Quantization\(^8\), in our experimentation. In the latter case, we have adopted the QS-PT (affine) scheme with histogram observer, considering the hardware practicality, for the activations. Whereas, for the weights, we have shown the results for both QS-PC (symmetric) with per-channel min-max observer & QS-PT (affine) with min-max observer, schemes. Observers record the range of the tensor and use it to compute the quantization parameters (\( \Delta, z \)). The details are beyond the scope of this work, hence, we refer the readers to\(^9\).

An overview of the proposed GZSQ framework has been given in the Algorithm 1.

IV. Experiments

This section first describes the details of the datasets, covering the vast diversity across the tasks, that have been used to evaluate the quantized models. It then shifts the attention towards the details of experimental settings and the state-of-the-art baselines, which have been appraised against the proposed GZSQ framework.

Datasets: We start our results (see: Section V) by showing the classification accuracy on CIFAR-10 [44] and ImageNet [10] datasets. CIFAR-10 consists of \( \sim 60K \) images across 10 different classes, of which pre-determined 10K samples have been used for testing the accuracy of the quantized model. ImageNet is one of the largest available benchmark datasets for image classification that consists of \( \sim 1.2M \) natural images for training and 50K for validation, across 1K classes. We have also presented the results for the task of object detection using Microsoft COCO [45] dataset. It consists of more than 200K images across 80 different object categories.

Lastly, to test the efficacy of the proposed GZSQ, we have evaluated the quantized models in the domain of medical imaging for the task of pneumonia classification in chest X-rays. The incorporated binary classification aims to categorize the given X-ray samples into normal and pneumatic cases. The pneumatic samples have been collected by utilizing both bacterial and viral infected cases. For this, we exploited the pre-split publicly available dataset\(^10\) consisting of \( \sim 6K \) chest X-rays. The complete dataset has been divided into train, validation, and test subsets. The training subset consists of 5.2K X-rays with 1341 and 3875 samples corresponding to normal and pneumatic cases. The validation subset comprises 16 equally distributed radiographs. Whereas the test set contains 624 images with 234 and 390 samples corresponding to normal and pneumatic cases. Samples of chest X-rays of normal and pneumatic persons are shown in Figure 1. The normal chest X-ray (see Figure 1; top) exhibits clear lungs without any abnormal opacified regions. Whereas the pneumatic X-ray (see Figure 1; bottom) typically depicts a focal lobar consolidation or interstitial pattern in one or both lungs.

It should be mentioned that for most of the tasks, we first adopt the publicly available pre-trained models, perform GZSQ, and test the quantized models on the validation or test set. For the tasks where it is difficult to get the pre-trained models, we first train the models using the suggested

\footnotesize{\(^{7}\)We use the quant. simulation of ZERQ + distillation process of GZSQ.}

\footnotesize{\(^{8}\)https://tutorials.pytorch.kr/advanced/static_quanitization_tutorial.html}

\footnotesize{\(^{9}\)https://pytorch.org/docs/stable/torch.quantization.html}

\footnotesize{\(^{10}\)https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia}
configuration in the respective papers. Later, we perform GZSQ and test the quantized models on the validation or test set. The proposed GZSQ scheme does not utilize either of the train, test, or validation samples in any form for any purpose at any stage to quantize a model.

**Experimental Environment:** We have performed our experiments using Pytorch [77] framework with Adam [78] optimizer and a learning-rate of $10^{-4}$ for 500 iterations on Nvidia Tesla P100 GPU. In all our experiments, mean results have been presented with the corresponding std-devs across 10 runs, each with a different random seed.

**Competing methods:** We have evaluated the proposed scheme on various models that are w/o BN [39] layers, such as ResNets [70], MobileNets [9], ShuffleNets [71], InceptionV3 [72], SqueezeNext [16] for the task of image classification. We have also considered the models that are w/o BN [39] layers, such as Fixup-ResNets [40] and ISONets [41], for image classification. RetinaNet [79] has been preferred for the task of object detection. For the case of medical imaging, we have considered the problem of Pneumonia classification in chest X-rays [76]. For this, we have adopted ResNet18 [70] and ResNext [80] models.

We have shown results against the recent best-published quantization works: RVQuant [81] (ECCV’18), DFQ [34] (ICCV’19), OCS [15] (ICML’19), ACIQ [31] (NIPS’19), GDFQ [38] (ECCV’20), DFC [36] (CVPR’20), and the most recent zero-shot work, namely, ZEROQ [35] (CVPR’20). For a fair comparison, all methods were evaluated using respective default parameters.

V. **Results**

**A. Classification**

Table II shows the CIFAR-10 classification accuracy on Fixup-ResNets [40] models that are w/o BN [39] layers, when quantized with different configuration. W8A8 denotes 8-bit integer precision for both weights and activations. It can be observed that for most of the Fixup-ResNet variants, the proposed GZSQ has outperformed the existing ZEROQ [35] approach across different quantization configurations, especially for W2A8. It should also be noted that for W2A8, even though the proposed GZSQ utilizes the pre-trained weights for estimating the $\Delta$ and $\sigma$ of the activations, there is not much degradation in the accuracy, compared to the ZEROQ [35].

Similarly, Table III shows the quantization accuracy of ISONets [41] models that are also w/o BN [39] layers on ImageNet [10] classification. It can be observed that for most of the models, GZSQ has achieved near FP32 accuracy, outperforming the ZEROQ [35] by a significant margin. From Tables II, III, it may be noted that ZEROQ [35] behaves similar to unit Gaussian in the absence of BN [39] layers due to its underlying backbone architecture, and results lower accuracy with quantized models. Whereas, the results obtained by using proposed GZSQ are not only superior to unit Gaussian and ZEROQ [35], but also close to FP32 accuracy.

Table IV shows the results obtained on ImageNet [10] classification for various models that consists of BN [39] layers. It can be observed that the proposed GZSQ has shown a remarkable improvement of over OCS [15] and ZEROQ [35] on majority of the models, especially with $+1.25\%$ for SqueezeNextV5 [16]. Tables, V, VI present the ImageNet [10] classification accuracy for MobileNetV2 [9] and ResNet18 [70] models, respectively. The results are categorized based on the methods that do not (a) No D: require original unlabelled samples, (b) No FT: require finetuning, (c) No AN: require additional network, and (d) No BN: require BN[39] layers in the network.

The results in Tables II:IV, XI are computed using ZEROQ’s [35] quantization simulation, whereas the results shown in Tables VII:IX, XIII are computed using Pytorch’s post-training static quantization framework. It can be observed from the Tables VII, VIII that the proposed GZSQ has outperformed the ZEROQ [35] on majority of the models. Even though we have assumed the input for the data distillation (see Sections III-A, III-B, III-C) to be a random $\mathcal{N}(0,1)$ instance, the GZSQ has shown a significant performance gain over Gaussian (0,1) input. It may be due to the incorporation of pre-trained weights in the proposed distillation process that has not been considered in ZEROQ [35].

It may also be concluded that the GZSQ’s distilled data has been more beneficial for inferring the activation ranges over $\mathcal{N}(0,1)$ and ZEROQ’s [35]. We have shown the results for both the cases when BN [39] folding is performed before and after data distillation. We have also shown the quantization results when a random subset of training set is used to infer the activation ranges.

**B. Object Detection**

Among high-level vision tasks, the complexity increases when attention shifts from classification to object detection in an image. Object detection aims to predict the set of bounding boxes around the objects with pre-defined category labels in an input image. To demonstrate the robustness of our approach, we have also evaluated the proposed scheme on the task of object detection using the Microsoft COCO [45] dataset. For this, we have adopted RetinaNet [79] single-stage detector which has a state-of-the-art results i.e., 36.4 mean average precision (mAP) (standard 0.5:0.05:0.95 metric). The adopted RetinaNet has a ResNet50 as backbone architecture. For quantization, in this paper, we have processed the pre-trained RetinaNet with INT8 configuration by using

---

11Considering the significant difference in the bit-precisions of 2-bits for weights and 8-bits for activations

12Available: https://pypi.org/project/pytorchcv/

13Source: https://github.com/pytorch/vision/tree/master/torchvision/models/quantization
TABLE II
TOP-1 ACCURACY ON FIXUP-RESNET [40] MODELS FOR THE TASK OF CIFAR-10 [44] CLASSIFICATION. QUANTIZATION CONFIGURATION IS WPtAQ
DENOTING THE P-BIT AND Q-BIT PRECISIONS USED FOR WEIGHTS (Q-P) AND ACTIVATIONS (Q-PT), RESPECTIVELY.

| Model (w/o BN) | W8A8 | W4A8 | W2A8 | FP32 |
|---------------|------|------|------|------|
| FixUp-ResNet 20 | 0.47±0.6e-3 | 0.49±0.6e-3 | 0.46±0.1e-3 | 0.48±0.6e-3 |
| FixUp-ResNet 32 | 0.22±0.1e-2 | 0.22±0.1e-2 | 0.23±0.1e-2 | 0.23±0.1e-2 |
| FixUp-ResNet 44 | 0.23±0.1e-2 | 0.23±0.1e-2 | 0.23±0.1e-2 | 0.23±0.1e-2 |
| FixUp-ResNet 56 | 0.33±0.1e-2 | 0.33±0.1e-2 | 0.33±0.1e-2 | 0.33±0.1e-2 |
| FixUp-ResNet 110 | 0.29±0.1e-2 | 0.29±0.1e-2 | 0.29±0.1e-2 | 0.29±0.1e-2 |

TABLE III
TOP-1 ACCURACY ON ISONET [41] MODELS FOR THE TASK OF IMAGE NET [10] CLASSIFICATION. “D” AND “NO-D” DENOTES WITH AND WITHOUT DROPOUT, RESPECTIVELY. QUANTIZATION CONFIGURATION IS INT8 WITH Q-P FOR WEIGHTS AND Q-PT FOR ACTIVATIONS.

| Model (w/o BN) | N(0.1) | ZERoQ [35] | GZSQ | FP32 |
|---------------|--------|------------|------|------|
| ISONet-18 | No-D | 6.58±0.10 | 6.56±0.11 | 6.76±0.02 | 6.94 |
| ISONet-34 | D | 6.87±0.10 | 6.87±0.10 | 7.10±0.00 | 7.45 |
| ISONet-50 | No-D | 6.98±0.10 | 6.96±0.00 | 7.05±0.00 | 7.38 |
| ISONet-101 | D | 6.98±0.10 | 6.96±0.00 | 7.05±0.00 | 7.38 |
| R-ISONet-18 | No-D | 6.74±0.10 | 6.73±0.00 | 7.01±0.00 | 7.29 |
| R-ISONet-34 | D | 6.96±0.10 | 6.96±0.00 | 7.01±0.00 | 7.29 |
| R-ISONet-50 | No-D | 7.06±0.10 | 7.05±0.00 | 7.11±0.00 | 7.31 |
| R-ISONet-101 | D | 7.33±0.10 | 7.32±0.00 | 7.42±0.00 | 7.54 |

TABLE IV
TOP-1 ACCURACY ON VARIOUS PRE-TRAINED MODELS FOR THE TASK OF IMAGE NET [10] CLASSIFICATION. QUANTIZATION CONFIGURATION IS INT8 WITH Q-P FOR WEIGHTS AND Q-PT FOR ACTIVATIONS.

| Model (w/o BN) | N(0.1) | ZERoQ [35] | GZSQ | FP32 |
|---------------|--------|------------|------|------|
| MNASNet-A0 | No-D | 6.58±0.10 | 6.56±0.11 | 6.76±0.02 | 6.94 |
| MNASNet-A2  | D | 6.87±0.10 | 6.87±0.10 | 7.10±0.00 | 7.45 |
| MNASNet-A4 | No-D | 6.98±0.10 | 6.96±0.00 | 7.05±0.00 | 7.38 |
| MNASNet-A6 | D | 6.98±0.10 | 6.96±0.00 | 7.05±0.00 | 7.38 |
| MobileNetV2 | No-D | 6.74±0.10 | 6.73±0.00 | 7.01±0.00 | 7.29 |
| MobileNetV2 | D | 6.96±0.10 | 6.96±0.00 | 7.01±0.00 | 7.29 |
| ResNet-50 | No-D | 7.06±0.10 | 7.05±0.00 | 7.11±0.00 | 7.31 |
| ResNet-101 | D | 7.33±0.10 | 7.32±0.00 | 7.42±0.00 | 7.54 |

C. Medical Imaging

Unlike natural images, X-rays and Magnetic Resonance Imaging (MRI) [84] scans do not comprise of many indistinguishable features, in general, among each other. It is challenging to differentiate between the X-rays or MRI scans of a normal and diseased person. Several methods based on deep learning frameworks have been proposed for many diseases [85, 86]. In this work, considering the COVID-19 [87] era, we have adopted one of the significant problems, namely, Pneumonia classification [88] in chest X-rays. For this, we have considered the ResNet18 [70] and ResNext101 [80] as baseline models and trained for the binary classification problem using the benchmark dataset [76] with a pre-determined train/val/test split. The trained model then has been quantized, and it can be observed from Table IX that the proposed GZSQ has outperformed the ZERoQ [35] and N'(0,1) by a noticeable margin.

D. Through the Lens of Class Activation Maps

Additional results are given in the provided supplementary material, wherein, Figure 5 shows the comparison among the class activation maps (CAMs) generated by utilizing the quantized models using ZERoQ [35], and GZSQ. It can be observed that the CAMs generated by using the GZSQ's quantized model are more similar to those of original FP32 model, than those of ZERoQ's. The readers are encouraged to refer Section VIII of the supplementary material for more details.

Table. X shows the reduction in size (MB) of the models due to quantization.

VI. ABLATION STUDY

In this section, first, we deeply investigate the distilled data for various models. We then present a brief ablation study of cost functions and the effect of extra biases introduced due to BN [39] folding before distillation.
TABLE VII

Top-1 accuracy on various models for the ImageNet [10] classification. Quantization configuration is INT8 with QS-PT for both weights and activations. * Used 1K samples (1/class) from ImageNet [10] training set for range calibration of the activations.

| Model (w BN) | N(0, 1) | ZEROQ [35] | GZSQ | Train* | FP32 |
|--------------|---------|------------|------|--------|------|
|              | Before  | After      | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  |
| ResNet18     | 67.42±0.23 | 67.78±0.26 | 67.50±0.21 | 15.97±1.16 | 69.29±0.05 | 69.09±0.05 | 69.37 | 69.29 | 69.76 |
| MobileNetV2  | 68.47±0.10 | 68.45±0.17 | 68.57±0.14 | 51.58±4.13 | 68.50±0.18 | 68.55±0.17 | 67.21 | 67.08 | 71.88 |
| ShuffleNetV2 | 62.47±0.13 | 62.38±0.11 | 62.34±0.20 | 58.27±1.11 | 64.04±0.12 | 64.02±0.18 | 64.74 | 64.13 | 69.36 |
| ResNet50     | 75.27±0.04 | 75.22±0.08 | 75.27±0.04 | 14.69±1.91 | 75.40±0.04 | 75.27±0.12 | 75.34 | 75.32 | 76.15 |
| InceptionV3  | 76.15±0.04 | 76.16±0.03 | 76.17±0.04 | 73.04±2.82 | 76.13±0.03 | 76.16±0.04 | 76.27 | 76.19 | 77.45 |

TABLE VIII

Top-1 accuracy on various models for the ImageNet [10] classification. Quantization configuration is INT8 with QS-PC for weights, and QS-PT for activations. * Used 1K samples (1/class) from ImageNet [10] training set for range calibration of the activations.

| Model (w BN) | N(0, 1) | ZEROQ [35] | GZSQ | Train* | FP32 |
|--------------|---------|------------|------|--------|------|
|              | Before  | After      | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  | Before | After  |
| ResNet18     | 67.80±0.37 | 67.78±0.19 | 67.76±0.17 | 16.17±2.13 | 69.47±0.05 | 69.33±0.06 | 69.43 | 69.44 | 69.76 |
| MobileNetV2  | 69.40±0.13 | 69.32±0.10 | 69.80±0.12 | 54.94±1.61 | 69.38±0.10 | 69.58±0.11 | 67.99 | 67.15 | 71.88 |
| ShuffleNetV2 | 67.23±0.13 | 67.25±0.17 | 67.51±0.13 | 58.60±2.21 | 67.72±0.08 | 67.88±0.14 | 68.15 | 68.17 | 69.36 |
| ResNet50     | 75.61±0.03 | 75.60±0.05 | 75.72±0.05 | 13.24±0.87 | 75.83±0.04 | 75.61±0.12 | 75.94 | 75.79 | 76.15 |
| InceptionV3  | 77.03±0.02 | 77.02±0.03 | 77.16±0.05 | 72.49±1.89 | 77.09±0.01 | 77.14±0.05 | 77.16 | 77.20 | 77.45 |

TABLE IX

Top-1 accuracy of the quantized models for Pneumonia classification in Chest X-Rays.

| Methods | ResNet18 | ResNet101_32x8d |
|---------|----------|----------------|
| N(0, 1) | Before   | 80.40±0.23 | 80.81±0.44 |
|         | After    | 80.36±0.12 | 80.89±0.47 |
| ZEROQ [35] | Before   | 80.40±0.14 | 81.04±0.35 |
|         | After    | 79.43±0.44 | 79.83±0.31 |
| GZSQ    | Before   | 80.56±0.36 | 81.16±0.53 |
|         | After    | 80.48±0.15 | 81.02±0.48 |
| FP32    |          | 80.80      | 81.20      |

TABLE X

Reduction in size (in MB) of the models due to quantization. M: Mean accuracy (INT8; QS-PT, before) on ImageNet [10] classification.

| Configs/Model | FP32 | INT8 | M-INT8 | M-FP32 |
|---------------|------|------|--------|--------|
| ResNet18      | 46.74 | 11.71 | 69.27 | 69.76 |
| MobileNetV2   | 13.98 | 8.58  | 68.50 | 71.88 |
| ShuffleNetV2  | 9.11  | 2.34  | 64.04 | 69.36 |
| ResNet50      | 102.14 | 25.05 | 75.40 | 76.15 |
| InceptionV3   | 95.32 | 23.95 | 76.15 | 77.45 |

A. Distilled Data

Figure 3 shows the histograms of distilled data obtained for the models that are w and w/o BN [39] layers. In Figure 3 (Row 1), the histograms of obtained distilled data have been plotted using unit Gaussian, ZEROQ [35] and GZSQ for the ISOnet [41] model (w/o BN layers). It can be observed that the ZEROQ [35], by its underlying backbone structure, results the distilled data similar to Gaussian but with a higher variance in the absence of BN [39] layers. This high-variance may be due to the initialization of distilled data using uniform distribution in ZEROQ [35] approach. The initialized uniform noise, in the absence of BN [39] layers, learns to shift towards a unit Gaussian within certain iterations. After finetuning, the left-out traces of uniform noise in the distilled data of the ZEROQ [35] may have caused the degradation in the accuracy for certain models, when compared to unit Gaussian and GZSQ, as shown in Tables. III, II.

Whereas, in Row 2, similar plots have been given for the MobileNetV2 [9] model (w BN [39] layers). To show the effect of BN [39] folding, which has been ignored in the ZEROQ [35], we have presented the plots for both the scenarios, when the BN [39] folding is performed before and after the data distillation. Unit Gaussian generated with different seeds is independent of the BN [39] folding. Therefore not much degradation has been observed in the case of N(0, 1), as shown in Tables. VII, VIII.

However, it can be observed from Figure 3 (Row 2) that distilled data obtained when BN [39] folding is performed before distillation is totally different when the same is performed after distillation. Especially, in the case of ZEROQ [35], which results in a significant difference between final accuracies (see Tables. VII, VIII). This is due to the fact that extra bias introduced because of BN [39] folding into the weights has not been considered in the ZEROQ’s distillation process. Consequently, it may have resulted in a poor correlation between the folded weights and the distilled data generated by looking at the BN [39] statistics prior to folding. Whereas, the same has been considered in the proposed GZSQ framework.

14We encourage the readers to refer this link for detailed analysis.
as described in Section III-D. Thus the obtained distilled data is not only distinguishable from unit Gaussian, but also does not exhibit any severe deviation because of the BN [39] folding. Figure 2 shows the histogram plots of the distilled data obtained for the task of pneumonia classification using ResNet101 [80].

B. Effect of cost functions

To show the effect of proposed $\mathcal{L}_Z$ cost function (see Eqn. 8), we have presented a detailed study by performing the following baselines:

- GZSQ-$\ell_1^p$, where $\ell_1$ norm between the means has been used as a cost function instead of proposed $\mathcal{L}_Z$,

$$ \|\mu_u - \mu_v\| $$

(13)

- GZSQ-$\ell_1^s$, where $\ell_1$ norm between the std-devs has been used as a cost function instead of proposed $\mathcal{L}_Z$,

$$ \|\sigma_u - \sigma_v\| $$

(14)

- GZSQ-$\ell_1$, where $\ell_1$ norm has been used as a cost function instead of proposed $\mathcal{L}_Z$,

$$ \|\mu_u - \mu_v\| + \|\sigma_u - \sigma_v\| $$

(15)

- GZSQ-$\ell_2^p$, where $\ell_2$ norm between the means has been used as a cost function instead of proposed $\mathcal{L}_Z$,

$$ \|\mu_u - \mu_v\|_2^2 $$

(16)

- GZSQ-$\ell_2^s$, where $\ell_2$ norm between the std-devs has been used as a cost function instead of proposed $\mathcal{L}_Z$,

$$ \|\sigma_u - \sigma_v\|_2^2 $$

(17)

- GZSQ-$\ell_2$, where $\ell_2$ norm (similar to ZEROQ [35]) has been used as a cost function instead of proposed $\mathcal{L}_Z$,

$$ \|\mu_u - \mu_v\|_2^2 + \|\sigma_u - \sigma_v\|_2^2 $$

(18)

- GZSQ-$\mathcal{L}_{KL}$, where KL divergence has been used to measure the difference between two distributions instead of proposed $\mathcal{L}_Z$. Since the estimated mean of BN [39] substitutes may have negative values (see Section III-A, ref: Eqn. 7), for which the KL divergence may be undefined. Hence, we have only utilized the std-devs to measure the difference. Therefore the adopted loss for GZSQ-$\mathcal{L}_{KL}$ is

$$ 1 - 0.5 \times KL(\sigma_u, \sigma_v), $$

(19)

where $KL(.)$ refers to Kullback-Leibler divergence, instead of Eqn. 8 [89].

It has been observed that for most of the cases, the parent baselines GZSQ-$\ell_1$ and GZSQ-$\ell_2$ have shown considerable superiority over aforementioned leaf baselines GZSQ-$\ell_1^p$, GZSQ-$\ell_1^s$, GZSQ-$\ell_2^p$, GZSQ-$\ell_2^s$, GZSQ-$\ell_2$. Further, it can be observed from Table. XI that the proposed $\mathcal{L}_Z$ cost function has been proven to be more beneficial than the $\ell_1$ and $\ell_2$ norms. It has also been observed that the $\ell_1$ and $\ell_2$ norms are useful in restoring the structural information (see Figure 4). Whereas the proposed $\mathcal{L}_Z$ loss restores the distributional properties of the data.
After GZSQ-
After GZSQ
Before
Before
.

TABLE XI
MEAN TOP-1 ACCURACY ON VARIOUS PRE-TRAINED MODELS FOR THE TASK OF IMAGE NET [10] CLASSIFICATION. QUANTIZATION CONFIGURATION IS INT8 WITH QS-PC FOR WEIGHTS AND QS-PT FOR ACTIVATIONS.

| Model (w BN) | GZSQ-\ell_2 | GZSQ-\ell_2 | GZSQ-\ell_2 | GZSQ-\ell_2 | GZSQ-\ell_2 | GZSQ-\ell_2 | GZSQ-\ell_2 | GZSQ-\ell_2 | GZSQ-\ell_2 | GZSQ-\ell_2 |
|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| ResNet18     | 69.09±0.05  | 69.09±0.05  | 69.09±0.05  | 69.09±0.05  | 69.09±0.05  | 69.09±0.05  | 69.09±0.05  | 69.09±0.05  | 69.09±0.05  | 69.09±0.05  |
| MobilenetV2  | 64.22±0.40  | 64.22±0.40  | 64.22±0.40  | 64.22±0.40  | 64.22±0.40  | 64.22±0.40  | 64.22±0.40  | 64.22±0.40  | 64.22±0.40  | 64.22±0.40  |
| ShuffleNetV2 | 61.00±0.14  | 61.00±0.14  | 61.00±0.14  | 61.00±0.14  | 61.00±0.14  | 61.00±0.14  | 61.00±0.14  | 61.00±0.14  | 61.00±0.14  | 61.00±0.14  |
| ResNet50     | 72.23±0.10  | 72.23±0.10  | 72.23±0.10  | 72.23±0.10  | 72.23±0.10  | 72.23±0.10  | 72.23±0.10  | 72.23±0.10  | 72.23±0.10  | 72.23±0.10  |
| InceptionV3  | 75.58±0.02  | 75.58±0.02  | 75.58±0.02  | 75.58±0.02  | 75.58±0.02  | 75.58±0.02  | 75.58±0.02  | 75.58±0.02  | 75.58±0.02  | 75.58±0.02  |

It can be observed that the ZERQ [35] retains some structural information when BN [39] folding is performed after distillation phase (see Figure 4, row:bottom, col:left). This may be due to the utilized BN [39] layer statistics. However, in this case, ZERQ [35] in general results the poor accuracy (see Section VI-A). Whereas, a significant loss in structural information can be observed when folding is performed prior to distillation (see Figure 4, row:top, col:left). Similar to ZERQ [35], we evaluated our baseline GZSQ-\ell_2 and observed that the proposed scheme is also able to retain the structural information when folding is performed after distillation phase (see Figure 4, row:top, col:middle). However, in contradiction to ZERQ [35], the proposed baseline also retains some structural information in the distilled data when folding is performed before distillation phase (see Figure 4, row:bottom, col:middle). It may be due to the incorporation of extra biases in the distillation phase, introduced due to BN [39] folding (see Section III-D), which may have carried, inherently, some information about the distribution of original training samples.

Therefore it can be concluded that some structural information about the data can be retained even without utilizing the BN [39] statistics in the distillation phase. However, such information may not be much beneficial for some models, e.g., SqueezeNextV5 [16], as shown in Table. XI or ZERQ’s [35] results for ResNet18 [70] as shown in Table. VII. Coming to the distributional minimization, we performed a baseline, namely, GZSQ-\ell_{KL}, that utilizes the KL divergence instead of the proposed \ell_Z cost function. However, it can observed from the Table. XII that the proposed \ell_Z loss yields better performance than the KL divergence based cost function. It should be mentioned that the KL divergence has been utilized with half of its capability since the mean values could not be incorporated in Eqn. 19. Though, it may be taken as a slight drawback of GZSQ. But, nothing for certain can be concluded about the end performance if KL divergence would have been utilized fully against the proposed \ell_Z.

It can also be observed from Figure 4 (row:top/bottom, col:right) that the stills of distilled data, when utilizing GZSQ, may not comprise of much structural information. However, it
TABLE XIII
| Model (w BN) | QS-PC | QS-PT | FP32 |
|-------------|-------|-------|------|
|             | w Bias | w/o Bias | w Bias | w/o Bias |
| ResNet18    | 69.47  | 69.27  | 69.76 | 69.76   |
| MobileNetV2  | 69.38  | 68.98  | 68.50 | 68.56   |
| ShuffleNetV2 | 67.72  | 67.78  | 64.04 | 65.97   |
| ResNet50    | 75.83  | 75.63  | 75.40 | 75.26   |
| InceptionV3 | 77.09  | 77.08  | 76.15 | 76.18   |

retains rich distributional properties that helps in achieving the near FP32 accuracy post quantization, as shown in Table. XI. When comparing with unit Gaussian samples, the inclusion of pre-trained weights in the distillation process of GZSQ may have acted as the cherry on the top.

C. Effect of BN folding

In Table XIII, we have shown the results to demonstrate the effect of extra biases, introduced due to BN[39] folding, when considered or ignored during our distillation process. It can be observed that for most of models, especially in the case of ResNet18 [70], incorporating extra biases in the distillation phase (see Section III-D) yields better results than ignoring the same in the distillation phase, irrespective of the quantization configuration. The incorporated biases may have reduced the correlation shift between the folded weights and distilled data, resulting in a near-accurate estimation of activation ranges to result in the same distributional properties.

VII. CONCLUSIONS

In this work, we have presented a novel generalized zero-shot quantization (GZSQ) framework for the post-training quantization of deep CNNs, that leverages only the pre-trained weights of the model. The proposed zero-shot approach does not rely on the original unlabelled data or the learned BN layer parameters to infer the activation ranges. Our proposed scheme is built upon the data distillation approach. And, we argued that Z-score based loss could be more effective than \( \ell_1 \) and \( \ell_2 \) norms or KL divergence, during distillation.

We have benchmarked the models (w & w/o BN layers) for the tasks of classification and object detection. We have also presented the quantization results for pneumonia classification in chest X-rays. We have shown that our method performed well, even when BN layer is folded or absent. Further, our approach is more generalized, outperforming the best-published works across multiple quantization frameworks. We have also presented a detailed ablation study, demonstrating the effect of various cost functions and BN folding.

The presented work is the first attempt towards the post-training zero-shot quantization of the futuristic unnormalized deep CNNs. And, in the subsequent time, we would like to extend our approach to mixed precision quantization and other problem domains. For e.g., multi-modal classification, image de-noising & reconstruction, a glimpse of whose foundation has been given in the Figure 6 of the supplemental file.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous Reviewers and Editors for their insightful observations, comments and suggestions, which helped to improve the quality of this work.

REFERENCES

[1] B. Brattoli, J. Tighe, F. Zhidanov, P. Perona, and K. Chalupka, “Re-thinking zero-shot video classification: End-to-end training for realistic applications,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2020.

[2] B. Xu, C. Lu, Y. Guo, and J. Wang, “Discriminative multi-modality speech recognition,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2020.

[3] X. Wang, Y. Liu, C. Shen, C. C. Ng, C. Luo, L. Jin, C. S. Chan, A. v. d. Hengel, and L. Wang, “On the general value of evidence, and bilingual scene-text visual question answering,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2020.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in International Conference on Learning Representations, 2015.

[7] A. Coates, B. Hsuval, T. Wang, D. J. Wu, A. Y. Ng, and B. Catanzaro, “Deep learning with large feature hierarchies,” in Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ser. ICML’13. JMLR.org, 2013, p. III–1337–1345.

[8] W. Gao, Y.-H. Liu, C. Wang, and S. Oh, “Rate distortion for model compression: From theory to practice,” in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2102–2111.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in CVPR’09, 2009.

[11] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao, “Hrank: Filter pruning using high-rank feature map,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[12] J. Gao, W. Ouyang, and D. Xu, “Multi-dimensional pruning: A unified framework for model compression,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[13] J.-H. Luo and J. Wu, “Neural network pruning with residual-connections and limited-data,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2020.

[14] Y. He, Y. Ding, P. Liu, L. Zhu, H. Zhang, and Y. Yang, “Learning filter pruning criteria for deep convolutional neural networks acceleration,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2020.

[15] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, and Z. Zhang, “Improving Neural Network Quantization without Retraining using Outlier Channel Splitting,” International Conference on Machine Learning (ICML), 2019.

[16] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, “Squeezennext: Hardware-aware neural network design,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2018.

[17] 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,” CoRR, vol. abs/1704.04861, 2017.

[18] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, and Z. Zhang, “Improving Neural Network Quantization without Retraining using Outlier Channel Splitting,” International Conference on Machine Learning (ICML), 2019.

[19] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, “Squeezennext: Hardware-aware neural network design,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2018.

[20] 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,” CoRR, vol. abs/1704.04861, 2017.
JOURNAL OF MRI, CLASS FILES, VOL. 14, NO. 8, AUGUST 2015

Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ser. ICCV '15. USA: IEEE Computer Society, 2015.

[70] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[71] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[72] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[73] P. Singh, V. K. Verma, P. Rai, and V. P. Namboodiri, “Leveraging filter correlations for deep model compression,” in 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), 2020.

[74] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size,” arXiv:1602.07360, 2016.

[75] S. Cao, L. Ma, W. Xiao, C. Zhang, Y. Liu, L. Zhang, L. Nie, and Z. Yang, “Searnet: Predicting convolutional neural network feature-map sparsity through low-bit quantization,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[76] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKenrow, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, “Identifying medical diagnoses and treatable diseases by image-based deep learning,” Cell, vol. 172, no. 5, pp. 1122 – 1131.e9, 2018.

[77] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alch´e-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[78] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd ICLR, San Diego, 2015.

[79] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in 2017 IEEE International Conference on Computer Vision (ICCV), 2017.

[80] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[81] E. Park, S. Yoo, and P. Vajda, “Value-aware quantization for training and inference of neural networks,” in Proceedings of the European Conference on Computer Vision (ECCV), September 2018.

[82] W. Sung, S. Shin, and K. Hwang, “Resiliency of deep neural networks under quantization,” CoRR, vol. abs/1511.06488, 2015.

[83] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, “Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems,” CoRR, vol. abs/1512.01274, 2015.

[84] D. W. McRobbie, E. A. Moore, M. J. Graves, and M. R. Prince, MRI from Picture to Proton, 2nd ed. Cambridge University Press, 2006.

[85] G. Yang, S. Yu, H. Dong, G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. Arridge, J. Keegan, Y. Guo, and D. Firmin, “Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction,” IEEE Transactions on Medical Imaging, 2018, 2018.

[86] O. Oktay, E. Ferrante, K. Kannitsas, M. Heinrich, W. Bai, J. Caballero, S. A. Cook, A. de Marvao, T. Dawes, D. P. O’Regan, B. Kaizn, B. Glock, and D. Rueckert, “Anatomically constrained neural networks (acnn): Application to cardiac image enhancement and segmentation,” IEEE Transactions on Medical Imaging, vol. 37, no. 2, 2018.

[87] A. Bagul, C. Langlotz, K. S. Shpanskaya, M. P. Lungren, and A. Y. Ng, “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning,” CoRR, vol. abs/1711.05225, 2017.

[88] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, “Dreaming to distill: Data-free knowledge transfer via deepinversion,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.
| PC    | Packet: 5.78 | Schooner: 5.55 | Schooner: 5.79 |
|-------|--------------|----------------|----------------|
| Schooner | | | |
| Zebra | Zebra: 9.31 | Zebra: 9.46 | Zebra: 9.62 |
| Warplane | Warplane: 9.06 | Warplane: 7.83 | Warplane: 8.52 |
| Aircraft | | | |
| Monkey | Monkey: 9.55 | Monkey: 9.82 | Monkey: 9.67 |
| Sandbar | Sandbar: 9.76 | Sandbar: 9.87 | Sandbar: 9.34 |
| Candle | Candle: 8.25 | Candle: 8.80 | Candle: 8.50 |
| Model T | Model T: 12.02 | Model T: 11.78 | Model T: 11.90 |
| Submarine | Submarine: 8.38 | Shp. Cart: 8.24 | Shp. Cart: 8.22 |

Fig. 5: Sample stills of class activation maps on the ImageNet [10] dataset using the quantized models from ZEROQ [35], and GZSQ. FP32 denotes the activation maps generated by using the original floating point model. We have utilized the SqueezeNextV5 [16] for this small study.
VIII. THROUGH THE LENS OF CLASS ACTIVATION MAPS (CONT.)

On an abstract level, class activation maps depict the regions in the input image, which have been addressed by the deep models with more focus during classification or any other similar high-level vision task. Figure 5 shows the class activation maps generated by using the quantized models for the task of ImageNet classification. For this, we have utilized the SqueezeNextV5 [16] model as FP32 baseline. This brief study aims to demonstrate the effect of quantization on the model’s focus through the lens of activation maps. In Figure 5, PC denotes the predicted class by the quantized model, whereas TC refers to the true class of the input image. It can be observed that across a variety of classes, activation maps generated by the quantized model using GZSQ are more similar to FP32 maps when compared to ZEROQ [35] maps. Further, the scores generated by the GZSQ quantized model are more aligned towards those of the FP32 against ZEROQ [35]. The warmer colors in the activation map denote the regions with more focus, whereas the lighter colors show the regions with less emphasis. Therefore, in terms of priority, it can be concluded that the quantized model using the GZSQ scheme performs more similarly to the original FP32 than the ZEROQ’s [35] quantized model.

IX. SINGLE IMAGE DE-RAINING USING QUANTIZED MODELS

TABLE XIV

| Metrics   | Behaviour | Input | N(0,1) | ZEROQ [35] | GZSQ | FP32 |
|-----------|-----------|-------|--------|------------|------|------|
| NIQE      |           | 4.7918 | 3.8296 | 3.8219     | 3.8162 | 3.6905 |
| BRISQUE   | L         | 32.0835 | 14.8261 | 15.1901    | 14.7678 | 13.3711|
| TV Error  | L         | 7.02×10⁶ | 6.48×10⁶ | 6.42×10⁶   | 6.50×10⁶ | 5.26×10⁶|

To show the robustness of the GZSQ across different domains, we have presented a brief ablation study on single image de-raining using quantized models. For this, we have adopted the pre-trained DRN [91] model as FP32 baseline and perform INT8 quantization using N(0,1), ZEROQ [35] and GZSQ. For quantitative analysis, we have utilized Real Internet [90] dataset that consists of 146 real-world rainy images collected from the internet. The selected real-world rainy images do not have ground truth; hence for a fair comparison, we have utilized the reference-less image quality metrics, namely, Naturalness Image Quality Evaluator (NIQE), Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE), and Total Variation Error (TV). The average spatial dimension of the images in the incorporated dataset is 371 px in height and 568 px in width. In general, image de-raining models trained on synthetic datasets tend to learn a different distribution of rain-streaks, unlike in real-world images, which is why most deep learning-based methods for image de-raining do not generalize well. However, in this paper, we restrict our analysis to the effect of quantization only. It can be observed from Table XIV that the results obtained by using GZSQ’s quantized model have surpassed ZEROQ [35] in terms of NIQE [92] and BRISQUE [93]. We have also presented the qualitative results in Figure 6 in terms of NIQE [92] against ZEROQ [35].

Fig. 6: Sample results on single image rain-streak removal by using the quantized models (INT8) from ZEROQ [35] and GZSQ in terms of Naturalness Image Quality Evaluator (NIQE; smaller is better). The original baseline FP32 model is DRN [91].