# Deep Transferring Quantization

Zheng Xie[1,2*], Zhiquan Wen[1*], Jing Liu[1*], Zhiqiang Liu[1,2], Xixian Wu[3], and Mingkui Tan[1†]

[1] South China University of Technology, Guangzhou, China
{sexiezheng,sewenzhiquan,seliujing,sezhiqiangliu}@mail.scut.edu.cn
mingkuitan@scut.edu.cn
[2] PengCheng Laboratory, Shenzhen, China
[3] HuNan Gmax Intelligent Technology, Changsha, China
wuxixian@gmax-ai.com

**Abstract** Network quantization is an effective method for network compression. Existing methods train a low-precision network by fine-tuning from a pre-trained model. However, training a low-precision network often requires large-scale labeled data to achieve superior performance. In many real-world scenarios, only limited labeled data are available due to expensive labeling costs or privacy protection. With limited training data, fine-tuning methods may suffer from the overfitting issue and substantial accuracy loss. To alleviate these issues, we introduce transfer learning into network quantization to obtain an accurate low-precision model. Specifically, we propose a method named deep transferring quantization (DTQ) to effectively exploit the knowledge in a pre-trained full-precision model. To this end, we propose a learnable attentive transfer module to identify the informative channels for alignment. In addition, we introduce the Kullback–Leibler (KL) divergence to further help train a low-precision model. Extensive experiments on both image classification and face recognition demonstrate the effectiveness of DTQ.

**Keywords:** Quantization · Deep Transfer · Knowledge Distillation

## 1 Introduction

Deep convolutional neural networks (CNNs) have been widely applied in various computer vision tasks, such as image classification [17, 18, 21, 48], face recognition [8, 12, 52], object detection [35, 41, 42], and semantic segmentation [7, 20, 44]. However, a deep model often contains millions of parameters and requires billions of floating-point operations (FLOPs) during inference, which restricts its applications on resource-limited devices, such as mobile phones. To reduce the computational costs and memory overheads, various studies have been proposed, such as low-rank decomposition [49, 63], network pruning [22, 37, 72] and network quantization [62, 68, 71]. In this paper, we focus on network quantization, which aims to compress the deep models and reduce the execution latency.

* Authors contributed equally.
† Corresponding author.

Existing quantization methods can be split into two categories, namely post-training quantization [2, 3, 67] and training-aware quantization [9, 28, 68]. Post-training quantization methods quantizes the models by directly converting weights and activations into low-precision ones. However, the performance will degrade severely in regard to low-precision quantization (e.g., 4-bit quantization). To achieve promising performance, training-aware quantization methods fine-tune the low-precision network with a large quantity of data to compensate for the performance loss from quantization. However, in many real-world scenarios, only a small number of labeled data are available due to expensive labeling costs or privacy protection. Since a deep model often contains a large number of parameters, fine-tuning with limited training data may easily suffer from the overfitting problem. Moreover, the training of a low-precision network is very challenging since the training process can easily get trapped in a poor local minimum, resulting in substantial accuracy loss [71]. This issue will be even more severe when the training data are limited.

To alleviate the data burden, we introduce transfer learning [6, 32, 39], which aims to transfer the knowledge from a source model to a target model. Transfer learning is an important machine learning paradigm that has several general characteristics: 1) the label space of the target task is different from that of the source task; 2) only a small quantity of labeled target data are available. Usually, we have a pre-trained full-precision model trained on the related source large-scale data set (e.g., ImageNet [45]), but the source data are often unavailable. The full-precision model contains rich and useful knowledge, which can be transferred to the low-precision model. Based on this intuition, we study the transferring quantization task, which seeks to obtain a promising low-precision model with a small number of target data while effectively exploiting the knowledge in the pre-trained full-precision model. Since training a low-precision model with limited target data is very challenging, we seek to propose a method to conduct network quantization and transferring simultaneously.

A deep model usually spans the data into a very high-dimensional space. Inspired by [59], the feature representations generated from a pre-trained model can be transferred to the target model. Imposing feature alignment between the two feature maps generated from the intermediate layer of the full-precision and low-precision models is a good way to exploit the knowledge. However, directly using feature alignment has a limitation. Due to the discrepancy between the target and source tasks, some channels of feature maps are irrelevant or even harmful for the discriminative power in the target task. In addition, such a phenomenon also exists between the full-precision and low-precision models. As a result, the low-precision model may have limited performance. Moreover, since the output of the full-precision model contains rich information about how the model discriminates an input image among a large number of classes, we can exploit this knowledge to guide the training of network quantization and improve the performance of the low-precision model under limited training data.

Based on the above intuition, we propose a simple but effective training method named deep transferring quantization (DTQ), which effectively exploits

the knowledge in a pre-trained full-precision model. To this end, we devise a learnable attentive transfer module to identify the informative channels and then align them generated from the full-precision and low-precision models for attentive transferring quantization (ATQ). Moreover, we propose probabilistic transferring quantization (PTQ) to force the probability distribution of the low-precision model to mimic that of the full-precision model.

Our main contributions are summarized as follows:

- In this paper, we study the task of transferring quantization, which introduces transfer learning into network quantization to obtain an accurate low-precision model. To our best knowledge, this task has not received enough attention from the community. Nevertheless, we argue and demonstrate that transferring quantization is very necessary when the target data are limited.
- We propose a simple but effective training method named deep transferring quantization (DTQ). This method uses attentive transferring quantization (ATQ) and probabilistic transferring quantization (PTQ) to effectively exploit the knowledge in the full-precision model for transferring quantization under limited training data. Extensive experiments on both image classification and face recognition demonstrate the superior performance of DTQ.

## 2 Related Work

**Network quantization** [19, 28] obtains a low-precision model that reduces model size and improves inference efficiency. Existing quantization methods can be divided into two categories, namely post-training quantization [2, 3, 67] and training-aware quantization [9, 28, 68]. Since post-training quantization does not require fine-tuning, the performance will degrade severely in regard to low-precision quantization (e.g., 4-bit). To compensate for the quantization performance decrease, training-aware quantization methods fine-tune the low-precision models with a large-scale training data set. Existing training-aware methods can be split into two categories: binary quantization [26, 40] and fixed-point quantization [9, 68]. For binary neural networks (BNNs), the weights and activations are constrained to $\{+1, -1\}$ [26, 40]. In this way, BNNs suffer from significant accuracy loss compared with full-precision models. To reduce this accuracy gap, fixed-point methods [9, 68] have been proposed to represent weights and activations with higher bitwidths. DoReFa-Net [68] designed quantizers with a constant quantization step and quantizes weights, activations and gradients to arbitrary bitwidths. Based on DoReFa-Net, PACT [9] used an activation clipping parameter that is optimized during the training to find the right quantization scale. Moreover, to achieve efficient integer-arithmetic-only inference, Jacob *et al.* [28] proposed a linear quantization scheme, which can be implemented more efficiently than floating-point inference on hardware. However, existing training-aware quantization methods require a large-scale labeled data set to conduct fine-tuning. In some cases, only limited training data are available, which limits the performance of the low-precision model.

**Transfer learning** [6, 39] seeks to transfer knowledge learned from the source data to the related target tasks. There are several works related to transfer learning, such as domain adaptation [46,64,65] and continual learning [30,34]. To explore the potential factors that affect deep transfer learning performance, Huh *et al.* [27] proposed analyzing features extracted by various networks pre-trained on ImageNet. Recently, several methods have been proposed to improve the transfer performance, such as sparse transfer [36], filter subset selection [10, 15] and parameter transfer [66]. Specifically, Li *et al.* proposed the learning without forgetting (LwF) approach [34], which used new target data to retrain models but preserved the knowledge of the source task. Motivated by LwF, Li *et al.* proposed $L^2$-$SP$ [33] to regularize the parameters between the two models, which forced the target model to approach the source model. However, if the regularization is too weak or too strong, it may hamper the generalization performance of the target model [32]. Recently, DELTA [32] proposed feature alignment at the channel level with channel attention. However, the attention is pre-learned and fixed for all samples. Since the true informative channels may vary for different samples, the shared attention may limit the transfer performance.

**Knowledge distillation [23]** (KD) is to distill the knowledge of a teacher network down to that of a small student network. The original works [1, 23] force the student networks to mimic the teacher networks to generate similar output distribution. Further, some methods [43, 72] proposed to align features in intermediate layers of two networks to transfer the knowledge. Recently, several methods [25, 58, 69] have been proposed to further exploit the knowledge of teacher networks. Specifically, Huang *et al.* formalized distillation as a distribution matching problem to optimize the student models [25]. Furthermore, many studies adopted the KD mechanism to train a compressed model for better performance. For example, Zhuang *et al.* [72] used KD to perform network pruning. Some network quantization methods [31, 53, 54, 71] proposed adopting KD to help train a low-precision model. However, these methods focus on knowledge transfer in the same tasks. In addition, the attention mechanism [24,38,55,56,60] for deep convoutional networks is relevant to this work.

## 3 Problem Definition

**Notation.** Throughout this paper, we use the following notations. Specifically, we use bold upper case letters (e.g., $\mathbf{W}$) to denote matrices and bold lower case letters (e.g., $\mathbf{x}$) to denote vectors. Let $M_{\text{full}}$ be a pre-trained full-precision model obtained on some related large-scale data sets (e.g., ImageNet) and $\mathbf{W}_{\text{full}}$ be the corresponding parameters. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ be the target training data, where $N$ is the number of samples. $M_{\text{low}}$ denotes a low-precision model and $\mathbf{W}_{\text{low}}$ denotes the corresponding parameters. Let $f(\mathbf{x}, \mathbf{W}_{\text{low}})$ be the prediction of $M_{\text{low}}$. Note that we may need to build a new classifier w.r.t. the target task by introducing a new fully-connected layer with new parameters $\mathbf{W}_{\text{low}}^{\text{FC}}$.

**Network quantization.** Given a pre-trained model, network quantization aims to reduce the model size and computational costs by mapping full-precision (i.e., 32-bit) weights and activations to low-precision ones. For each CNN layer, quantization is parameterized by the number of quantization levels and clamping range. Considering $k$-bit linear quantization [28], the quantization function is:

$$\begin{aligned} &\text{clamp}(r; a, b) = \min(\max(r, a), b)\,, \\ &s(a, b, k) = \frac{b-a}{2^k - 1}\,, \\ &q = \text{round}(\frac{\text{clamp}(r; a, b) - a}{s(a, b, k)})s(a, b, k) + a\,, \end{aligned} \tag{1}$$

where $r$ denotes the full-precision value, $q$ denotes the quantized value, $[a, b]$ is the quantization range, $2^k$ is the number of quantization levels, and the round$(\cdot)$ function denotes rounding to the nearest integer.

**Transferring quantization.** In many practical scenarios, only limited labeled data are available. In this case, existing quantization methods directly fine-tune a low-precision model, which may easily suffer from the overfitting issue. Moreover, network quantization transforms the continuous values into discrete values, leading to the worse representational ability of the network. Hence, the low-precision training process can easily get trapped in a poor local minimum, resulting in substantial performance degradation. Usually, we have a pre-trained full-precision model $M_{\text{full}}$, which is obtained on some large-scale data sets (e.g., ImageNet) and contains rich knowledge. Hence, effectively exploiting the knowledge of the full-precision model will help the training of quantization. Based on this intuition, we study a task named **transferring quantization**, which aims to obtain a promising low-precision model by effectively exploiting $M_{\text{full}}$ and the limited data in $\mathcal{D}$. Note that the source data are often unavailable.

## 4 Proposed Method

To effectively exploit the knowledge in the full-precision model $M_{\text{full}}$, one feasible method imposes feature alignment on the intermediate feature maps between $M_{\text{full}}$ and $M_{\text{low}}$. In this way, the knowledge in $M_{\text{full}}$ is expected to be transferred into $M_{\text{low}}$. Since the target task is often different from the source task, some channels of feature maps are irrelevant or even harmful for the discriminative power in the target task. Moreover, such a phenomenon also exists between the full-precision model $M_{\text{full}}$ and the low-precision model $M_{\text{low}}$. Thus, directly applying feature alignment may not obtain promising performance. In addition, since the output of the full-precision model contains rich knowledge about discriminating information among a large number of classes [23], the output probability distribution of the full-precision model $M_{\text{full}}$ can be also regarded as a guided signal for training the low-precision model $M_{\text{low}}$. Based on this above intuition, in the following, we propose a simple but effective training method

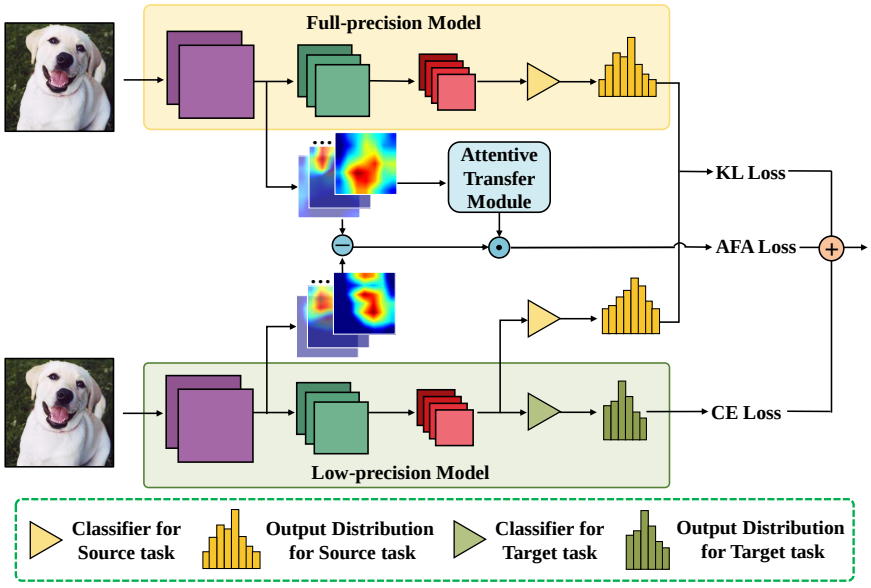


**Figure 1.** An overview of DTQ. KL loss is the Kullback–Leibler divergence loss, AFA loss is the attentive feature alignment loss and CE loss is the cross-entropy loss. Note that we evenly take feature maps from four intermediate layers for alignment

named deep transferring quantization (DTQ), which simultaneously performs network quantization and knowledge transfer.

### 4.1 Deep Transferring Quantization

Motivated by the attention mechanism [55] and knowledge distillation (KD) [43, 60], to effectively exploit the useful knowledge in a pre-trained full-precision model, we first devise a learnable attentive transfer module to identify the informative channels. As mentioned above, it is important to exclude the irrelevant channels and focus on the informative channels for attentive transferring quantization (ATQ). Second, we introduce the Kullback–Leibler (KL) divergence to measure the discrepancy of the probability distribution between the full-precision model and the low-precision model. By minimizing the KL divergence, the useful knowledge learned from the source data can be transferred to the low-precision model for probabilistic transferring quantization (PTQ).

Let $\mathrm{P}_{\mathrm{full}}$ and $\mathrm{P}_{\mathrm{low}}$ be the full-precision model and low-precision model predictions for the source task, respectively. With the introduction of ATQ and PTQ, we perform transferring quantization by minimizing this objective w.r.t. $\mathbf{W}_{\mathrm{low}}$ and attention parameters $\mathbf{W}_a$:

$$\sum_{i=1}^{N}(\mathcal{L}(f(\mathbf{x}_i,\mathbf{W}_{\mathrm{low}}),y_i)+\alpha\Omega(\mathbf{W}_{\mathrm{full}},\mathbf{W}_{\mathrm{low}},\mathbf{W}_a,\mathbf{x}_i))+\beta D_{KL}(\mathrm{P}_{\mathrm{full}}||\mathrm{P}_{\mathrm{low}})\,, \quad (2)$$

**Algorithm 1** Deep Transferring Quantization

**Input**: A pre-trained full-precision model $M_{\text{full}}$, target training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, the number of epochs $T$, the batch size $m$, and the hyperparameters $\alpha$ and $\beta$.
**Output**: A low-precision model $M_{\text{low}}$.

1: Initialize $M_{\text{low}}$ based on $M_{\text{full}}$.
2: **for** $t = 1, \ldots, T$ **do**
3: Randomly sample a mini-batch $(\mathbf{x}, y) \sim \mathcal{D}_m$.
4: Update $\mathbf{W}_{\text{low}}$ and $\mathbf{W}_a$ by minimizing Eq. (2).
5: **end for**

where $\mathcal{L}$ refers to the empirical loss (e.g., cross-entropy loss), $\Omega$ denotes the attentive feature alignment (AFA) loss, $D_{KL}$ is the KL divergence loss, and $\alpha$ and $\beta$ are trade-off hyperparameters. In this way, we can effectively exploit the useful knowledge in the full-precision model $M_{\text{full}}$ to obtain a promising low-precision model $M_{\text{low}}$. An overview of DTQ is shown in Fig. 1, and the overall algorithm is shown in Algorithm 1.

### 4.2 Attentive Transferring Quantization

In this subsection, we introduce attentive transferring quantization (ATQ) in detail. The feature maps derived from the pre-trained full-precision model may contain irrelevant or even detrimental channels to the target low-precision model. To alleviate this issue, we devise an attentive transfer module (ATM) to focus on the discriminative channels of feature maps. As shown in Fig. 2, ATM adopts a two-layer perceptron MLP with a softmax layer to recognize the informative channels. Based on ATM, the attention weight vector $\mathbf{a}^i$ for the $i$-th sample can be formulated as

$$\mathbf{a}^i = \text{Softmax}(\text{MLP}(\mathbf{W}_a, g(\text{F}(\mathbf{W}_{\text{full}}, \mathbf{x}_i)))) \,, \tag{3}$$

where $\text{F}(\cdot,\cdot)$ denotes a feature map; $g(\cdot)$: $\mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{C\times HW}$ flattens the feature maps in spatial dimension; $H$, $W$, $C$ are the height, width and number of channels of the feature maps, respectively; $\mathbf{W}_a$ denotes the parameters of MLP. Based on the attention weight vector, ATQ focuses on the informative channels by minimizing the attentive feature alignment (AFA) loss:

$$\Omega(\mathbf{W}_{\text{full}}, \mathbf{W}_{\text{low}}, \mathbf{W}_a, \mathbf{x}_i) = \sum_{l=1}^{L}\sum_{j=1}^{C} \text{a}_j^i \|\text{F}_{l_j}(\mathbf{W}_{\text{low}}, \mathbf{x}_i) - \text{F}_{l_j}(\mathbf{W}_{\text{full}}, \mathbf{x}_i)\|_{\text{F}}^2 \,, \tag{4}$$

where $L$ is the number of intermediate layers for alignment. Note that DELTA [32] adopted a similar attention mechanism to ATQ, but the attention mechanism of DELTA is pre-learned and then fixed for all samples when training a target model. In our method, each sample has a unique attention weight vector. Last, our method simultaneously updates $\mathbf{W}_{\text{low}}$ and $\mathbf{W}_a$.

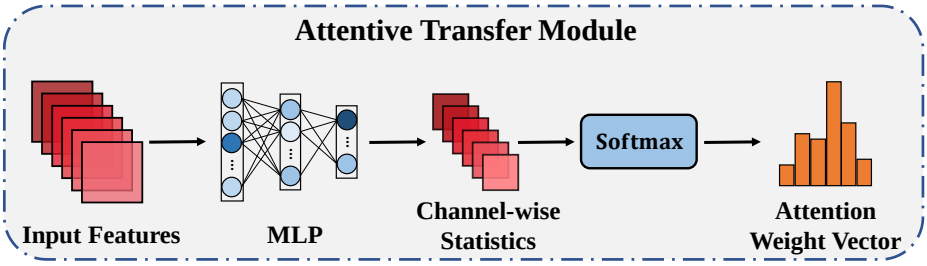


**Figure 2.** An overview of our attentive transfer module. For each sample, input features first enter a two-layer perceptron MLP to obtain channel-wise statistics. Then, channel-wise statistics pass the softmax layer to obtain the attention weight vector

### 4.3 Probabilistic Transferring Quantization

In this subsection, we introduce probabilistic transferring quantization (PTQ) in detail. Except for using attentive feature alignment for ATQ, we also force the probability distribution of the low-precision model to mimic that of the full-precision model by PTQ. However, the number of classes in the target task is often different from that in the source task. It is impossible to directly apply the Kullback–Leibler (KL) divergence to the output of two models. To solve this, we reuse the classifier of the full-precision model on the low-precision model to obtain the probability distribution regarding the source task, as shown in Fig. 1.

Similar to [23, 70], to measure the correlation between the two probability distributions, i.e., $\mathrm{P}_{\mathrm{full}}$ and $\mathrm{P}_{\mathrm{low}}$, we employ the KL divergence:

$$D_{KL}(\mathrm{P}_{\mathrm{full}} \,||\, \mathrm{P}_{\mathrm{low}}) = \sum_{i=1}^{N} \mathrm{P}_{\mathrm{full}}(\mathbf{x}_i) \log \frac{\mathrm{P}_{\mathrm{full}}(\mathbf{x}_i)}{\mathrm{P}_{\mathrm{low}}(\mathbf{x}_i)} \,. \tag{5}$$

By minimizing the KL divergence between the two probability distributions, the knowledge in the output distribution can be transferred from the full-precision model to the low-precision model.

## 5 Experiments on Image Classification

### 5.1 Source and Target Data Sets

We choose a large-scale image classification data set as the source data set, namely, ImageNet [11]. The small-scale target data sets are five public data sets with different domains: Stanford Dogs 120 [29], Food-101 [4], CUB-200-2011 [51], Caltech 256-30 and Caltech 256-60 [16]. Similar to [33], we consider Caltech 256-$x$, where $x$ denotes the number of samples for each class for training (e.g., Caltech 256-10). For validation, we randomly sample 20 images for each class. We show the details of these target data sets in Table 1.

**Table 1.** Characteristics of the target data sets: name, number of classes, and number of samples of the training set and validation set

| Target Data Sets | # Classes | # Training | # Validation |
|---|---|---|---|
| Stanford Dogs 120 | 120 | 12,000 | 8,580 |
| Caltech 256-30 | 257 | 7,710 | 5,140 |
| Caltech 256-60 | 257 | 15,420 | 5,140 |
| CUB-200-2011 | 200 | 5,994 | 5,794 |
| Food-101 | 101 | 75,750 | 25,250 |

### 5.2 Compared Methods

To our best knowledge, the task of transferring quantization has not received enough attention from the community and we fail to find very related baselines for comparison. To evaluate the proposed DTQ, we construct the following methods for comparison: ***L*²-Q** [4]: directly fine-tune all the parameters of the low-precision model with weight decay on the target data. ***L*²-*SP*-Q**: based on $L^2$-$SP$ [33], we regularize the parameters between two models as a part of the loss function to encourage the low-precision model to be similar to the full-precision model. **DELTA-Q**: relying on transferring quantization, we follow DELTA [32] to align feature maps between the full-precision model and the low-precision model with a fixed channel attention mechanism.

### 5.3 Implementation Details

We adopt MobileNetV2 [47] and ResNet-50 [21] as base models, and use the pre-trained full-precision models from torchvision. We use SGD with a mini-batch size of 64, where the momentum term is set to 0.9. The initial learning rate is set to 0.01. We train low-precision models for 9k iterations, and the learning rate is divided by 10 at the 6k-th iteration. For the hyperparameters $\alpha$ and $\beta$ in Eq. (2), we fix $\beta$ to 0.5 and use cross-validation to search for the best $\alpha$ for each experiment. Following in [5], in Eq. (1), we set $a$ and $b$ to the minimum and maximum of the values, respectively. Following [32], we take feature maps from four intermediate layers for attentive feature alignment (i.e., $L = 4$ in Eq. (4)). We repeat each experiment five times, and report the average Top-1 accuracy and the standard deviation on the validation set. The source code and the pre-trained models are available at https://github.com/xiezheng-cs/DTQ.

### 5.4 Results and Discussions

First, we directly fine-tune the low-precision network on the target data set with different quantization methods, including DoReFa [68], PACT [9] and linear quantization [28]. For convenience, we use $L^2$-X to denote that directly fine-tune with X quantization method. From Table 2, compared with $L^2$-DoReFa and

[4] We follow the naming rule from the transfer learning community to name methods.

**Table 2.** Comparisons of different methods. We quantize MobileNetV2 and report the Top-1 accuracy (%) on five target data sets. "W" and "A" represent the quantization bitwidth of the weights and activations, respectively

| Target Data Sets | W / A | $L^2$-DoReFa | $L^2$-PACT | $L^2$-Q | $L^2$-$SP$-Q | DELTA-Q | DTQ |
|---|---|---|---|---|---|---|---|
| Stanford Dogs 120 | 5 / 5 | 48.4±1.2 | 48.1±1.0 | 73.3±0.2 | 75.2±0.1 | 79.3±0.2 | **80.2±0.2** |
| | 4 / 4 | 48.5±1.2 | 48.2±1.0 | 69.1±0.3 | 70.9±0.4 | 76.0±0.3 | **76.1±0.4** |
| Caltech 256-30 | 5 / 5 | 45.6±1.5 | 46.0±1.4 | 74.9±0.3 | 75.6±0.2 | 78.3±0.3 | **80.1±0.2** |
| | 4 / 4 | 44.9±1.0 | 44.8±0.4 | 68.2±0.3 | 69.2±1.3 | 74.9±0.7 | **75.9±1.2** |
| Caltech 256-60 | 5 / 5 | 58.7±1.3 | 58.5±1.4 | 78.3±0.1 | 79.4±0.2 | 82.4±0.3 | **83.2±0.2** |
| | 4 / 4 | 58.1±0.8 | 58.2±0.7 | 73.6±0.4 | 74.1±0.2 | 79.2±0.6 | **79.9±0.3** |
| CUB-200-2011 | 5 / 5 | 47.3±1.4 | 46.0±1.5 | 75.3±0.2 | 75.1±0.2 | **76.2±0.3** | 75.4±0.3 |
| | 4 / 4 | 46.3±1.5 | 47.3±0.9 | 69.2±0.9 | 70.0±1.0 | 71.9±0.2 | **72.1±0.3** |
| Food-101 | 5 / 5 | 66.3±1.0 | 66.7±0.8 | 81.3±0.1 | 81.1±0.3 | **81.7±0.1** | **81.7±0.1** |
| | 4 / 4 | 64.6±0.7 | 64.9±0.7 | 77.4±0.2 | 76.9±0.4 | 77.7±0.5 | **78.4±0.4** |

**Table 3.** Comparisons of different methods. We quantize ResNet-50 and report the Top-1 accuracy (%) on five target data sets. "W" and "A" represent the quantization bitwidth of the weights and activations, respectively

| Target Data Sets | W / A | $L^2$-Q | $L^2$-$SP$-Q | DELTA-Q | DTQ |
|---|---|---|---|---|---|
| Stanford Dogs 120 | 5 / 5 | 78.9±0.3 | 84.3±0.3 | 86.5±0.5 | **86.5±0.4** |
| | 4 / 4 | 75.2±0.2 | 80.8±0.2 | 81.7±2.0 | **82.3±0.7** |
| Caltech 256-30 | 5 / 5 | 83.3±0.2 | 83.3±0.1 | 85.0±0.2 | **85.0±0.1** |
| | 4 / 4 | 78.9±0.2 | 80.6±0.1 | 82.8±0.5 | **83.5±0.6** |
| Caltech 256-60 | 5 / 5 | 84.4±0.3 | 86.6±0.2 | 87.2±0.1 | **87.4±0.2** |
| | 4 / 4 | 80.8±0.7 | 84.3±0.3 | 84.8±1.1 | **85.5±0.8** |
| CUB-200-2011 | 5 / 5 | 80.3±0.1 | 80.2±0.3 | **80.9±0.2** | 80.3±0.1 |
| | 4 / 4 | 76.3±0.2 | 76.6±0.2 | 76.4±0.3 | **77.8±0.3** |
| Food-101 | 5 / 5 | 84.1±0.1 | 84.4±0.2 | 84.3±0.1 | **84.4±0.1** |
| | 4 / 4 | 80.5±0.3 | 80.6±0.2 | 80.6±0.7 | **81.0±1.5** |

$L^2$-PACT, $L^2$-Q achieves much better performance. This indicates that linear quantization is more suitable for transferring quantization under the limited training data. Thus, we adopt linear quantization as our quantization method.

Second, we compare the performance of DTQ with three methods, including $L^2$-Q, $L^2$-$SP$-Q and DELTA-Q. We show the results of MobileNetV2 and ResNet-50 in Table 2 and Table 3, respectively. From these results, we make the following observations. 1) For the 5-bit MobileNetV2 and ResNet-50, DTQ outperforms these compared methods in most cases. 2) DTQ achieves significant improvement over the baselines, especially at low-precision (e.g., 4-bit) quantization. Specifically, for 4-bit MobileNetV2, DTQ outperforms DELTA-Q in the Top-1 accuracy by 1.0% on Caltech 256-30; for 4-bit ResNet-50, DTQ outperforms these compared methods in the Top-1 accuracy by at least 0.4% on all target data sets. Moreover, DTQ surpasses DELTA-Q in the Top-1 accuracy by 1.4% on CUB-200-2011. In a word, these results show the effectiveness of DTQ.

**Table 4.** Effect of different losses in DTQ. We report the Top-1 accuracy (%) of 5-bit MobileNetV2 on the Caltech 256-30 data set

| Model | CE Loss | AFA Loss | KL Loss | Top-1 Accuracy |
|---|---|---|---|---|
| MobileNetV2 (5-bit) | √ | | | 74.9±0.3 |
| | √ | √ | | 79.7±0.1 |
| | √ | √ | √ | **80.1±0.2** |

### 5.5 Effect of Losses in DTQ

To investigate the effect of the losses in DTQ, we first conduct experiments with different combinations of the losses for the 5-bit MobileNetV2 on Caltech 256-30. Then, we visualize the feature maps of the models with different losses in DTQ. The experimental results are shown in Table 4 and Fig. 3.

**Quantitative comparisons.** From the results of 5-bit MobileNetV2 in Table 4, we make the following observations. 1) CE Loss: this baseline just uses the cross-entropy (CE) loss to train a low-precision network. 2) CE Loss + AFA Loss: this experiment increases the Top-1 accuracy by about 5.0% compared with the method just using CE loss. Besides, DTQ achieves 79.7% in Top-1 accuracy, which is significantly better than DELTA-Q (78.3% in Table 2). These results embody the effectiveness of our proposed AFA loss. 3) CE Loss + AFA Loss + KL Loss: compared with the second experiment, it further improves the performance of the low-precision network. These results indicate the effectiveness of our KL loss. In total, both AFA loss and KL loss contribute to better performance of the low-precision models.

**Qualitative comparisons.** To further investigate the effect of the losses in DTQ, we visualize the feature maps of the penultimate layer of MobileNetV2 on Caltech 256-30. From Fig. 3, when we only use the CE loss, the 5-bit MobileNetV2 fails to focus on the target object. When we add the AFA loss, the 5-bit MobileNetV2 achieves significantly better performance, whose feature maps activate the information of the target object more accurately. Furthermore, the 5-bit MobileNetV2 equipped with all three losses shows a better concentration on the target object than that equipped with two losses. Due to the page limit, we put more visualization results in the supplementary. This visualization results further demonstrate the effectiveness of the proposed losses in DTQ.

### 5.6 Further Experiments

**Performance on different scales of data sets.** We conduct several experiments to evaluate the proposed DTQ on different scales of data sets. We choose Caltech 256 with different numbers of training samples for each class, i.e., from 10 to 60. From the results of 4-bit MobileNetV2 in Fig. 4, DTQ outperforms other methods on different scales of Caltech 256, especially on the small scale

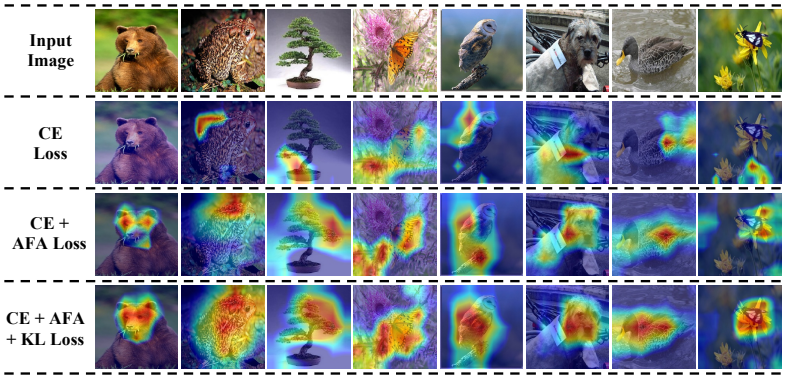


**Figure 3.** Visualization of features from models with different losses. Samples are taken from the features of the penultimate layer of 5-bit MobileNetV2 on Caltech 256-30

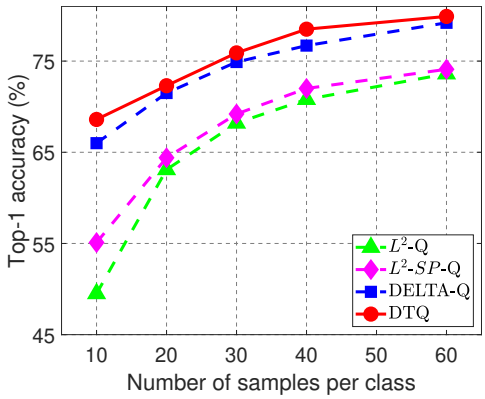


**Figure 4.** Performance of different methods in the Top-1 accuracy (%) of 4-bit MobileNetV2 on different scales of the Caltech 256 data set

of data set. For example, on the training data set with 10 training samples for each class, DTQ outperforms DELTA-Q by 2.6% in the Top-1 accuracy. These results indicate the superiority of DTQ under small training data.

**Effect of the attentive transfer module.** To evaluate the proposed attentive transfer module (ATM), we conduct experiments on the DTQ with and without attention. DTQ without attention means directly using feature alignment without ATM for ATQ. From the results in Table 5, DTQ with attention outperforms the one without attention, especially on Food-101 (1.3% improvement on average Top-1 accuracy), which demonstrates the effectiveness of our ATM.

**Effect of different training schemes.** To further investigate the effect of training scheme, we extend DTQ to two-stages training scheme. Specifically, we

**Table 5.** Effect of different training schemes. "One-stage" refers to performing transferring and quantization simultaneously. "Two-stage" denotes that we first perform transferring and then perform quantization. We report the Top-1 accuracy (%) of 4-bit MobileNetV2 on five target data sets. Note that "w/o ATT" means without attention

| Target Data Sets | W / A | One-stage | | Two-stage | |
|---|---|---|---|---|---|
| | | DTQ (w/o ATT) | DTQ | DT→ $L^2$-Q | DT → DTQ |
| Stanford Dogs 120 | 4 / 4 | 75.2±0.5 | 76.1±0.4 | 71.5±0.4 | **76.1±0.5** |
| Caltech 256-30 | 4 / 4 | 75.1±0.4 | 75.9±1.2 | 74.2±0.2 | **76.7±0.9** |
| Caltech 256-60 | 4 / 4 | 79.1±0.4 | 79.9±0.3 | 77.2±0.1 | **80.9±0.3** |
| CUB-200-2011 | 4 / 4 | 71.6±0.3 | 72.1±0.3 | 70.7±0.1 | **73.3±0.2** |
| Food-101 | 4 / 4 | 77.1±0.7 | 78.4±0.4 | 77.5±0.4 | **79.9±0.4** |

**Table 6.** Performance comparisons of different methods on the PolyU-NIRFD data set. "FAR" denotes the false acceptance rate

| LResNet18E-IR (4 bit) | True Acceptance Rate (%) | | |
|---|---|---|---|
| | FAR=1e-5 | FAR=1e-6 | FAR=1e-7 |
| $L^2$-Q | 84.9 | 79.9 | 75.9 |
| $L^2$-$SP$-Q | 89.1 | 85.8 | 83.1 |
| DELTA-Q | 91.8 | 89.9 | 88.7 |
| DTQ | **93.3** | **91.1** | **90.3** |

do transferring in the first stage and perform quantization in the second stage. Let "DT" be conducting transferring without quantization by using the DTQ framework. We consider the following methods for comparison. **DT → $L^2$-Q** : we apply DT in the first stage to obtain a full-precision model and then apply $L^2$-Q to train a low-precision model in the second stage. **DT → DTQ** : we apply DT in the first stage and DTQ in the second stage. Note that the pre-trained full-precision model does not change during the training process.

We quantize MobileNetV2 with different methods on five target data sets. From the results in Table 5, compared with two-stage DT → $L^2$-Q, one-stage DTQ outperforms it by a large margin. For example, one-stage DTQ surpasses two-stage DT → $L^2$-Q by 4.5% in Top-1 accuracy on Stanford Dogs 120. These results, to some extent, imply the necessity to perform transferring and quantization simultaneously. Furthermore, two-stage DT → DTQ outperforms one-stage DTQ on five target data sets. These results demonstrate that a better initialized point in DTQ achieves better performance.

## 6 Experiments on Face Recognition

In this experiment, we evaluate the proposed DTQ on the face recognition task. We use the visible light (VIS) face data set CASIA-WebFace [57] as the source data set, and the PolyU near-infrared ray (NIR) face data set (PolyU-NIRFD) [61] as the target data set. Besides, we adopt LResNet18E-IR [13] as

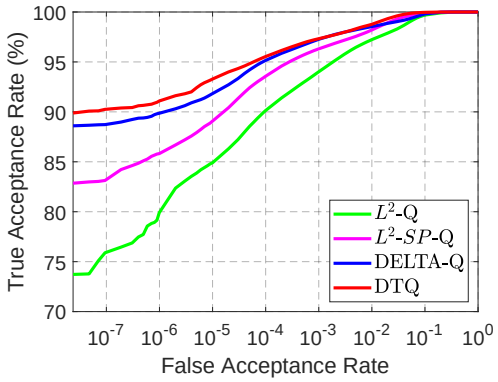


**Figure 5.** ROC [14,50] curves of 4-bit LResNet18E-IR on the PolyU-NIRFD data set

the base model. We report the resutls in Table 6 and Fig. 5. From the results, our DTQ achieves the best performance. Due to the page limit, we put more implementation details and results in the supplementary. These results demonstrate the effectiveness of the proposed DTQ on the face recognition task.

## 7 Conclusion

In this paper, we have studied the transferring quantization task, which aims to obtain a promising low-precision model by effectively exploiting the pre-trained full-precision model with limited training data. To achieve accurate low-precision models, we have proposed a simple but effective method named deep transferring quantization (DTQ). In our DTQ, we devised an attentive transfer module to identify informative channels, and further proposed attentive transferring quantization (ATQ) to align the informative channels of the low-precision model with that of the full-precision model. In addition, we introduced the Kullback–Leibler (KL) divergence on the probability distribution of two models for probabilistic transferring quantization (PTQ). By minimizing the KL divergence, the useful knowledge learned from the source data can be transferred to the low-precision model. Extensive experimental results on both image classification and face recognition demonstrate the effectiveness of the proposed DTQ.

## Acknowledgements

This work was partially supported by the Key-Area Research and Development Program of Guangdong Province 2019B010155002, National Natural Science Foundation of China (NSFC) 61836003 (key project), Program for Guangdong Introducing Innovative and Entrepreneurial Teams 2017ZT07X183, Fundamental Research Funds for the Central Universities D2191240.

## References


1. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: Conference on Neural Information Processing Systems. pp. 2654–2662 (2014)
2. Banner, R., Nahshan, Y., Hoffer, E., Soudry, D.: ACIQ: Analytical clipping for integer quantization of neural networks. arXiv preprint arXiv:1810.05723 (2018)
3. Banner, R., Nahshan, Y., Soudry, D.: Post training 4-bit quantization of convolutional networks for rapid-deployment. In: Conference on Neural Information Processing Systems. pp. 7948–7956 (2019)
4. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: European Conference on Computer Vision. pp. 446–461 (2014)
5. Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: ZeroQ: A novel zero shot quantization framework. arXiv preprint arXiv:2001.00281 (2020)
6. Caruana, R.: Multitask learning. Machine learning **28**(1), 41–75 (1997)
7. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (2017)
8. Chen, S., Liu, Y., Gao, X., Han, Z.: MobileFaceNets: Efficient cnns for accurate real-time face verification on mobile devices. In: Chinese Conference on Biometric Recognition. pp. 428–438 (2018)
9. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.J., Srinivasan, V., Gopalakrishnan, K.: PACT: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085 (2018)
10. Cui, Y., Song, Y., Sun, C., Howard, A., Belongie, S.: Large scale fine-grained categorization and domain-specific transfer learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
11. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.: ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
12. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: Additive angular margin loss for deep face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)
13. Deng, J., Guo, J., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. arXiv preprint arXiv:1801.07698v1 (2018)
14. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters **27**(8), 861–874 (2006)
15. Ge, W., Yu, Y.: Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1086–1095 (2017)
16. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset (2007)
17. Guo, Y., Chen, Y., Zheng, Y., Zhao, P., Chen, J., Huang, J., Tan, M.: Breaking the curse of space explosion: Towards efficient nas with curriculum search. In: International Conference on Machine Learning (2020)
18. Guo, Y., Zheng, Y., Tan, M., Chen, Q., Chen, J., Zhao, P., Huang, J.: NAT: Neural architecture transformer for accurate and compact architectures. In: Conference on Neural Information Processing Systems. pp. 735–747 (2019)
19. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In: International Conference on Learning Representations (2016)

20. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: IEEE International Conference on Computer Vision. pp. 2980–2988 (2017)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
22. He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: IEEE International Conference on Computer Vision. pp. 1389–1397 (2017)
23. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv:1503.02531 (2015)
24. Hong, S., Oh, J., Lee, H., Han, B.: Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3204–3212 (2016)
25. Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219 (2017)
26. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks. In: Conference on Neural Information Processing Systems. pp. 4107–4115 (2016)
27. Huh, M., Agrawal, P., Efros, A.A.: What makes imagenet good for transfer learning? arXiv:1608.08614 (2016)
28. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2704–2713 (2018)
29. Khosla, A., Jayadevaprakash, N., Yao, B., Li, F.F.: Novel dataset for fine-grained image categorization: Stanford dogs. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops. vol. 2 (2011)
30. Kirkpatrick, J., Pascanu, R., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017)
31. Leroux, S., Vankeirsbilck, B., Verbelen, T., Simoens, P., Dhoedt, B.: Training binary neural networks with knowledge transfer. Neurocomputing **396**, 534–541 (2020)
32. Li, X., Xiong, H., et al.: DELTA: Deep learning transfer using feature map with attention for convolutional networks. In: International Conference on Learning Representations (2019)
33. Li, X., Grandvalet, Y., Davoine, F.: Explicit inductive bias for transfer learning with convolutional networks. In: International Conference on Machine Learning. pp. 2830–2839 (2018)
34. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(12), 2935–2947 (2017)
35. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: IEEE International Conference on Computer Vision. pp. 2980–2988 (2017)
36. Liu, J., et al.: Sparse deep transfer learning for convolutional neural network. In: AAAI Conference on Artificial Intelligence (2017)
37. Luo, J.H., Wu, J., Lin, W.: ThiNet: A filter level pruning method for deep neural network compression. In: IEEE International Conference on Computer Vision. pp. 5058–5066 (2017)

38. Moon, S., Carbonell, J.G.: Completely heterogeneous transfer learning with attention-what and what not to transfer. In: International Joint Conferences on Artificial Intelligence (2017)
39. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2009)
40. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: XNOR-Net: Imagenet classification using binary convolutional neural networks. In: European Conference on Computer Vision. pp. 525–542 (2016)
41. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
42. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Conference on Neural Information Processing Systems. pp. 91–99 (2015)
43. Romero, A., Ballas, N., et al.: FitNets: Hints for thin deep nets. In: International Conference on Learning Representations (2015)
44. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 234–241 (2015)
45. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
46. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: European Conference on Computer Vision. pp. 213–226 (2010)
47. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
48. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
49. Tai, C., Xiao, T., Wang, X., E, W.: Convolutional neural networks with low-rank regularization. In: International Conference on Learning Representations (2016)
50. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: DeepFace: Closing the gap to human-level performance in face verification. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1701–1708. IEEE Computer Society (2014)
51. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
52. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: CosFace: Large margin cosine loss for deep face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5265–5274 (2018)
53. Wei, Y., Pan, X., Qin, H., Ouyang, W., Yan, J.: Quantization mimic: Towards very tiny cnn for object detection. In: European Conference on Computer Vision. pp. 267–283 (2018)
54. Xu, J., Nie, Y., Wang, P., López, A.M.: Training a binary weight object detector by knowledge transfer for autonomous driving. In: International Conference on Robotics and Automation. pp. 2379–2384 (2019)
55. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International Conference on Machine Learning (2015)

56. Yao, Y., Ren, J., Xie, X., Liu, W., Liu, Y., Wang, J.: Attention-aware multi-stroke style transfer. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1467–1475 (2019)
57. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
58. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
59. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Conference on Neural Information Processing Systems. pp. 3320–3328 (2014)
60. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: International Conference on Learning Representations (2017)
61. Zhang, B., Zhang, L., Zhang, D., Shen, L.: Directional binary code with application to polyu near-infrared face database. Pattern Recognition Letters (2010)
62. Zhang, D., Yang, J., Ye, D., Hua, G.: LQ-Nets: Learned quantization for highly accurate and compact deep neural networks. In: European Conference on Computer Vision. pp. 365–382 (2018)
63. Zhang, X., Zou, J., He, K., Sun, J.: Accelerating very deep convolutional networks for classification and detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(10), 1943–1955 (2015)
64. Zhang, Y., Chen, H., Wei, Y., et al.: From whole slide imaging to microscopy: Deep microscopy adaptation network for histopathology cancer image classification. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 360–368 (2019)
65. Zhang, Y., Wei, Y., Zhao, P., et al.: Collaborative unsupervised domain adaptation for medical image diagnosis. IEEE Transactions on Image Processing (2020)
66. Zhang, Y., Zhang, Y., Yang, Q.: Parameter transfer unit for deep neural networks. In: The Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 82–95 (2019)
67. Zhao, R., Hu, Y., Dotzel, J., De Sa, C., Zhang, Z.: Improving neural network quantization without retraining using outlier channel splitting. In: International Conference on Machine Learning. pp. 7543–7552 (2019)
68. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016)
69. Zhu, M., Wang, N., Gao, X., Li, J., Li, Z.: Face photo-sketch synthesis via knowledge transfer. In: International Joint Conferences on Artificial Intelligence. pp. 1048–1054 (2019)
70. Zhuang, B., Liu, J., Tan, M., Liu, L., Reid, I., Shen, C.: Effective training of convolutional neural networks with low-bitwidth weights and activations. arXiv preprint arXiv:1908.04680 (2019)
71. Zhuang, B., Shen, C., Tan, M., Liu, L., Reid, I.: Towards effective low-bitwidth convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 7920–7928 (2018)
72. Zhuang, Z., Tan, M., Zhuang, B., Liu, J., Guo, Y., Wu, Q., Huang, J., Zhu, J.: Discrimination-aware channel pruning for deep neural networks. In: Conference on Neural Information Processing Systems. pp. 875–886 (2018)