Quantization Table Redesign for the JPEG Compression Algorithm Targeting Classification Neural Networks

Abstract

Image compression is an important technique in deep learning applications as it effectively reduces the required storage size, lowers data transfer overhead and shortens classification time. JPEG is a widely used lossy image compression method involving a quantization table made of a 8×8 array, which largely influences the quality as well as the degree of the JPEG compression. The existing research on generating quantization table using human visual system approach, rate-distortion approach, and meta-Heuristics approach are not optimal for deep neural networks. In this work, we primarily focus on exploring efficient approaches to produce a neural network favorable JPEG quantization table. We use hyper-parameter tuning methods including random search, sorted random, bounded random search, Bayesian optimization and multi-armed bandit to redesign a quantization table based on data retrieved from the 'ImageNetV2' dataset. The new quantization tables we obtained can provide performance improvement by 20% to 200% increase in compression rate when the accuracy is fixed, or up to 2% at the same compression rate. With cross validation on different datasets and retrained neural network, the improvement is decreased but not disappeared.

1 Introduction

Neural network used to be a major area of research for neuroscience nad computer science till 1969. In the next few decades, this technique enjoyed a resurgence in natural language processing, self-driving cars, speech recognition and object detection (Collobert & Weston, 2008; Hirschberg & Manning, 2015; Bojarski et al., 2016; Abdel-Hamid et al., 2014; Szegedy et al., 2013). Recently, with the emergence of mobile devices, internet of things and cloud storage services, neural network are gaining unprecedented growth and are one of the commonly used machine learning algorithms for clustering and classification now. This brings the issue of managing the storage or memory requirement for a dataset to its practitioners. If a model can achieve equivalent performance at higher compression ration, acquisition, transmission and storage of large dataset can be less prohibitive.

As one of the most commonly used image compression standards for neural network datasets including ImageNet (Deng et al., 2009), PASCAL VOC (Everingham et al.) COCO (Lin et al., 2014), and etc., JPEG (Joint Photographic Experts Group) standard is susceptible to quality distortion (Dodge & Karam, 2016). Fig. 1 shows an overview of the standard JPEG compression algorithm. Each digital component, i.e. the YCbCr channels, are first partitioned into 8×8 non-overlapping blocks. Then the digital components are transformed to frequency components with 2D discrete cosine transform (DCTII) transforms. At quantization, the frequency components are quantized by a scaled quantization table. The scale is determined by quality factor ranges from 0 to 100 (LuaDist, 2015). For instance, a quality factor of 100 scales quantization table coefficients into 0 and the frequency components are rounded into integer. Then the quantized coefficients are ordered into the "Zig-Zag" sequence and further compressed with entropy coding and Huffman coding.

Among all steps, quantization is particularly interesting and challenging since it is lossy and the given quantization table decides what features are preserved, potentially playing a critical role for neural network classification when the dataset is compressed by JPEG. However, existing researches on designing quantization table mostly prioritize human perceived distortions as compression target (Liu et al., 2018), which compressed images that may not be distinguishable to computers (Wright et al., 2009). Other work that addresses this issue such as DeepN-JPEG (Liu et al., 2018) validates its quantization table using ImageNet, a dataset already compressed by JPEG.

In this work, we discuss traditional JPEG quantization table design targeting PSNR and human visual system unfit for deep neural network in Section 2. Section 3 shows how we rebuild the a high resolution with 1933 × 1592 pixels on average by retrieving images from Flickr with ID provided by ImageNetV2. To find better quantization table, we start with sorted random search, and based on the bound we obtain with sorted random search, we further test different hyper-parameter tuning methods, including sorted random search, bound random search, Bayesian optimization, and MAB, to

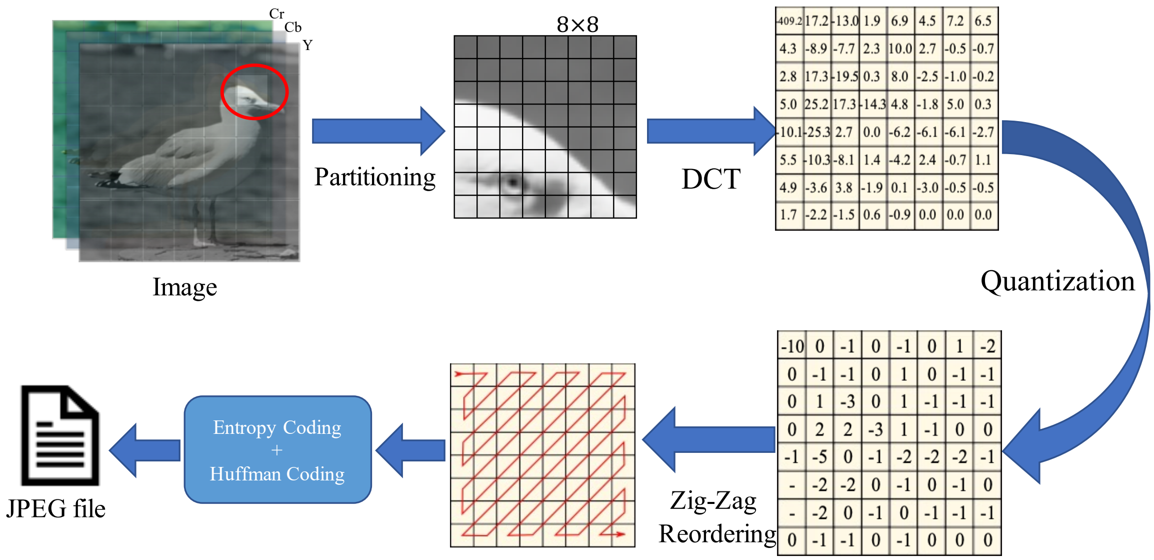

Figure 1. Overview of JPEG Compression Algorithm

design a neural network favorable JPEG quantization table in Section 4. We compare the performance and training efficiency of all different methods, cross-validates them on other parts of ImageNetV2 and ImageNet in Section 5.

The contributions of this paper are:

- We apply sorted random search to all range of compression rates and show that standard quantization table can be easily outperformed by this simple algorithm.
- With bounds set by sorted random search, we compare the performance of different methods and analyze their efficiency.
- We demonstrate the effectiveness of our methods by cross-validating the pareto optimal quantization tables with different testing datasets and through retraining.
- We build a high-resolution dataset based on ImageNetV2 and compress it to mimic the process of dataset compression.

Our experiments show that given same compression rate the testing accuracy can be improved by 1% to 2% through quantization table redesign. This corresponds to 20% to 200% of improvement on compression rate given the same testing accuracy. The accuracy improvement can retain even after cross validation on different datasets and retraining neural network.

2 Related Work and Motivation

2.1 Quantization Table Optimization

Because JPEG is one of the most widely used image compression techniques, JPEG quantization table optimization has been an enduring issue. The existing approaches can be classified into three categories: rate-distortion approach, human visual system approach, and meta-heuristic approach (Naresh et al., 2015). Rate-distortion approaches introduce distortion, measured by peak signal-to-noise ratio (PSNR), and gradually increases compression rate (Wu & Gersho, 1993; Fung & Parker, 1995; Ramchandran & Vetterli, 1994). Progressively, human visual system approach became popular, which aiming at optimizing visual quality for an image (Watson, 1993; Wang et al., 2001; Westen et al., 1996; Jiang & Pattichis, 2011). Standard JPEG is designed under the same goal (Wallace, 1992). However, neither PSNR or human visual system models target this issue from the perspective of deep neural network. Instead, it is designed to cater human visual system which is not sensitive to many small details as the computers do. What's more, existing popular meta-heuristic approaches for quantization table optimization requires large population to start with (Wu, 2004; Costa & Veiga, 2005; Cutello et al., 2005; Balasubramanian & Manavalan, 2016; Ma & Zhang, 2013; Tušar & Filipič, 2007; Kumar & Karpagam, 2016; Tuba & Bacanin, 2014) However, compressing all images as input data consumes a long time. In our experiments, generating a population of 300 dataset with 2500 images takes a day on a 20 threading Intel(R) Xeon(R) CPU E5-2620 v4 CPU. We therefore aim at finding efficient methods in search for the optimal space.

2.2 DeepN-JPEG

There are few work that highlight the optimization of quantization table for neural network. DeepN-JPEG models the difference between human visual system and deep neural network. Using the difference, they design their quantization table structure according to the frequency bands of DCT components. However, this quantization table structure may be limiting its performance for neural networks. Instead, we leverage on hyper-parameter tuning to guide us in finding a better quantization table for neural network itself.

Also, the authors both determine parameters of their quantization table and test it on ImageNet, a pre-compressed JPEG-format dataset of low resolution. In this work, we rebuild a high-resolution dataset from ImageNetV2 and cross validate all kinds of datasets to learn the generality of the quantization tables we find.

3 Dataset

ImageNet is already downsized and lossy compressed compressed. For instance, ImageNet 2013 classification dataset has an average resolution of 482×415 pixels (Russakovsky et al., 2015). Instead, we turn to ImageNetV2, a new test dataset for ImageNet (Recht et al., 2019). In ImageNetV2, the images are also downsized and compressed, but the authors open-source their code to generate the dataset as well as the IDs of which we can retrieve the source image with Flickr API (Wikipedia contributors, 2019). The dataset we rebuild has an average resolution of 1933×1592 pixels.

ImageNetV2 contains three test sets, breaking into Matched-Frequency, TopImages and Threshold0.7 based on the sampling strategies (Recht et al., 2019). Each test set has 1000 classes with 10 images per class. Specifically, we use the MatchedFrequency test set of which was sampled to match the MTurk selection frequency distribution of the original ImageNet validation set for each class. At training, we use a small portion of MatchedFrequency dataset to cross validate our result and also speedup the training process.

4 Methods

We treat the whole problem as hyper-parameter tuning and does not retrain the neural network but only feed in input images compressed by different quantization table. Since compressing the dataset for one validation is expensive, our methods aim at finding optimal points as few trials as possible and each trial shall not take long decision time. We used sorted random and bounded random search, Bayesian optimization and MAB, to find quantization tables that outperforms the standard one.

4.1 Sorted Random Search

A simple and yet effective method is random search. However, a complete random search results in search space of $256^{64} \approx 1.34 \cdot 10^{154}$. Random sampling would thus not generate a good search result. Instead, when we observe a typical output of DCT coefficients, the upper left ones are large while the lower right ones are relatively small. Typical 8×8 DCT coefficients can be found at the third step in Fig. 1. We therefore perform a Zig-Zag reordering on a sorted list of 64 integers. The integers are uniformly random generated among $[s, e]$ where $s, e \in \mathbb{Z}$ and $1 \leq s < e \leq 255$. The intuition also comes from JPEG compression algorithm where after quantization, it performs a Zig-Zag reordering for further compression. We refer to this Zig-Zag reordering method as sorted random search.

The method creates a pareto curve in terms of accuracy and compression rate and we denote the set of points on that curve as PA. Each point $p \in PA$ has three fields, accuracy acc, compression rate cr and quantization table terms $t[i]$ $i \in [0, 63]$. This experiment provides a good starting point for other algorithms.

4.2 Bounded Random Search

Previously, it is hard to capture good points by randomly sampling from a tremendously large search space. With sorted random search, we obtain a bound in the search space where we can now extensively explore. Bounds for a large range of compression rate is also large, making it difficult for sampling, while we only care the most interesting point of quality 50, where the scaling factor of quantization table is 100%. The compression rate of the standard JPEG quantization table is approximately 22 at quality 50. We take points from the pareto set PA with compression rate range around 22 to from a new set $PA_{quality} \subseteq PA$, where $21 \leq p.cr \leq 23 \; \forall p \in PA_{quality}$. The boundary for each term $t[i], i = 0, 1, \ldots 63$ in quantization table is set to

$$\begin{aligned} &\forall p \in PA_{quality}, \\ &\text{LowerBound}[i] = \min(p.t[i], p.t[63-i]) - 0.5\sigma(p.t[i]), \\ &\text{UpperBound}[i] = \max(p.t[i], p.t[63-i]) + 0.5\sigma(p.t[i]). \end{aligned}$$

By randomly sampling within the bounds, we avoided the constraint of ordering.

4.3 Bayesian Optimization

Bayesian optimization is widely used for hyper-parameter searching in neural network and can efficiently find good points in search space. To take this advantage of Bayesian optimization, we need to address two problems.

First, Bayesian optimization is modeled towards a single target objective function and our task involves two: compression rate and accuracy. Given certain compression rate, even when there is no corresponding point in the pareto set PA, we can use a parabola $fitness(CR) = aCR^2 + bCR + c$ that fits the pareto curve to estimate best accuracy we can achieve with sorted random search. Then we can set the target value $y = ACC - fit(CR)$, standing for the difference between the accuracy gap between actual accuracy and best accuracy found by sorted random search. The large search space is another issue. With it being as large as $1.34 \cdot 10^{154}$, any sampling algorithms have negligible chance of sampling good points. We therefore set the sampling bound the same as bounded random search. Also, low frequency

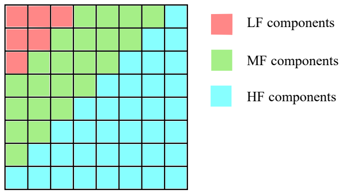

Figure 2. Frequency Band Partitions.

(LF) and middle frequency (MF) components of DCT 8x8 arrays have larger absolute value (Kaur et al., 2011), and therefore more susceptible to changes of quantization value. We define the area of interest as low frequency and middle frequency bands as shown in Fig 2. We perform a local grid search to randomly choose indexes in the area of interested indexes and iteratively update the quantization table with the highest acquisition value.

```
x_samples = generate_uniform(n_init)
ys = acquisition(x_samples)
max_acq = ys.max()
x_max = x_samples[ys.argmax()]
for i in range(n_greedy_updates):
  indexes = random_choose(
    n_choices, interest_indexes)
  x_samples = grid_search(
    indexes, bounds)
  ys = acquisition(x_samples)
  if (ys.max() > max_acq):
    max_acq = ys.max()
    x_max = x_samples[ys.argmax()]
```

4.4 Multi-Arm Bandit

Multi-Arm Bandit (MAB) is one of the most fundamental techniques in Reinforcement Learning (RL). The main idea of MAB is to learn a strategy to interact with the unknown environment and maximize the expected reward from the environment. Similar to Bayesian optimization, MAB maintains a set of probabilistic distribution estimate for each bandit arm, and will update these distributions according to history results. In each iteration, the RL agent calculates the score of each arm (i.e. options available in the decision making) based on the estimate, and returns the most promising arm (with the highest score), which will be validated in the next run. The score function for arm a at iteration t is defined as followed:

$$U_t(a) = \bar{x}_a + \sqrt{\frac{-log(t)}{2N_t(a)}} * c \tag{1}$$

The x_a is the mean reward of arm a from the search history and $N_t(a)$ denotes how many times arm a has been selected before iteration t. Factor c is the weight factor to balance exploration and exploitation. It is set as 0.05 by default, and we can modify its value to adjust the balance between exploration and exploitation. The searching is more aggressive (i.e. more efforts in exploration) with a larger weight factor.

There are different ways to formalize the quantization table searching problem into MAB. For example, we can partition the incredibly large search space into sub-spaces, and consider each subspace as an arm. In this section, we consider heuristic algorithms as bandit arms to avoid the limitation of different heuristics (e.g. some heuristics are only effective to limited set of problems). We developed the MAB technique based on OpenTuner (Ansel et al., 2014), which is an extensive, generic auto-tuning framework, and selected particle swarm optimization (PSO),simulated annealing (SA), differential evolution (DE), Greedy Mutation and Random Nelder Mead as the bandit arms. All these heuristics are generic optimization algorithms and can be performed without much computation overhead.

To avoid being stuck in the local minimal, we introduced some randomness for decision making process. i.e., there is small probability that the low-score arms will be selected in each run. The MAB algorithm will select the most promising heuristic in each iteration, and allocate computation resource effectively to strike a good balance between exploration and exploitation.

5 Experiments and Results

The models are evaluated on 500 classes with 5 images each of MatchedFrequency dataset to speedup compression and cross validation. The model we use is ResNet50 **??** provided by PyTorch API (Paszke et al., 2017). Our searching is based on top-1 accuracy since it is hard to weigh multiple accuracy goals, and the compression rate is calculated as the ratio of BMP images to compressed JPEG images.

Sorted random search is the starting point for us to treat quantization table coefficients as hyper-parameters. Because bounded random search, Bayesian optimization and MAB are built on top of boundary information gained from sorted random search, we discuss them in one subsection. We also compare the efficiency of different searching technique and how they generalize to other datasets in terms of both testing and training.

5.1 Sorted Random Search

Fig. 3 is the result from 4000 sorted random search and 1000 random search, with JPEG quantization table quality set from 10 to 100 at an interval of 5. Through random search, we did not find any quantization table that outperforms the standard JPEG quantization table. Sorted random search found pareto points from 1% to 2% given the same compression rate when the quality factor of standard JPEG is in the range of $[15, 90]$. The points where quality factor are 5 and 95 the quantization table coefficients are close to

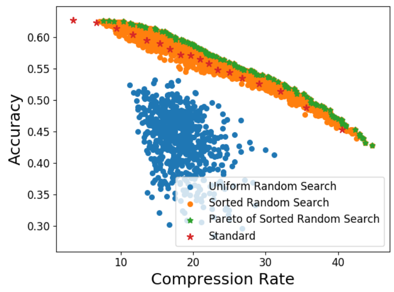

Figure 3. Sorted Random Search and Random Search Performances Compared to Standard JPEG

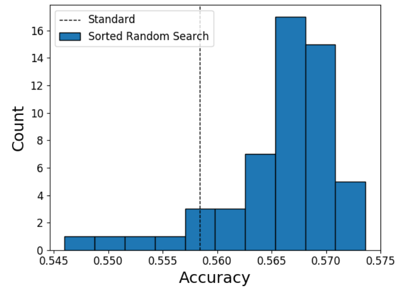

Figure 4. Sorted Random Search Distribution with Compression Rate Range $(22, 22.2)$

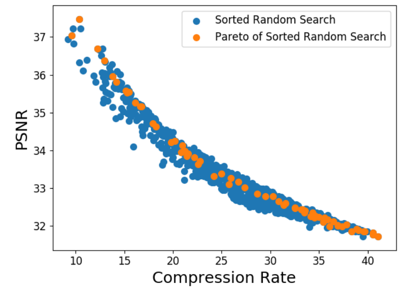

Figure 5. PSNR of Sorted Random Search with Pareto of Compression Rate and Accuracy Highlighted

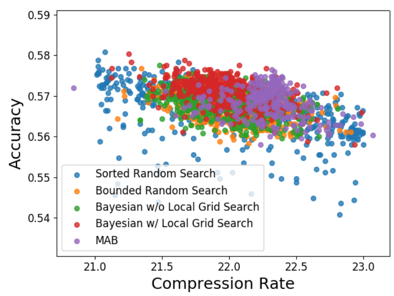

Figure 6. Bounded Searching Methods

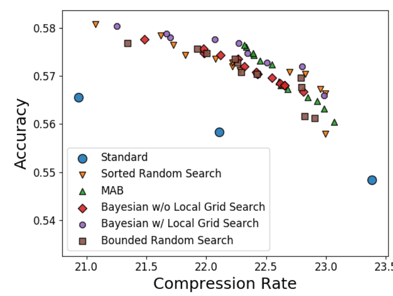

Figure 7. Pareto of Bounded Searching Methods

1 and 255 and therefore provide limited space for optimization. Given the same accuracy, sorted random search can find quantization tables with compression rate 20% to 200% higher than the standard JPEG depending on the baseline compression rate.

Fig. 4 shows the distribution of sorted random points with compression rate between 22 to 22.2. Among 52 points, there are 48 points perform at least as good as the JPEG standard quantization table, of which 93.75% perform better.Taking into account the strong randomness of sorted random search, the result indicates JPEG standard quantization table is far from optimal for deep neural networks.

To demonstrate the difference between PSNR and deep neural network optimization goal, we highlight the pareto in terms of compression rate and accuracy in Fig. 5, showing the PSNR and compression rate of sorted random search points. Fig. 5 shows the PSNR of sorted random search points and we highlight the pareto in terms of compression rate and accuracy. Though PSNR is closely related to neural network accuracy, it can not be taken as the sole indicator of high accuracy.

5.2 Comparison among Bounded Searching Methods

Due to the time constraint of this course project,we have 618 points from bounded random search, 610 points from Bayesian optimization without local grid search, 416 points from Bayesian optimization with local grid search and 262 points from MAB. Fig. 6 shows the scatter points of all bounded searching algorithms. We also plot the points from sorted random search in the range of $(21, 23)$ as baseline. The experiments indicate that setting bounds for the searching range can regulate the target compression rate. Simple technique like bounded random search can give in a better distribution than sorted random search. It might because there is no forced structure of bounded random search. Bayesian optimization with local grid search and MAB are two methods with highest mean and smallest standard distribution. Figure 7 shows the pareto points of all methods in comparison to standard JPEG quantization perform better. Bayesian optimization and MAB can provide best quantization tables among all methods. Bayesian optimization without local grid search does not show a strong advantage over sorted and bounded random search in terms of finding pareto optimums mainly because the sampling space is too large to sample a good point.

5.3 Efficiency Analysis

We also profiled the efficiency of different methods. All the experiments are built on a 2 thread per core, 8 cores per socket and 2 sockets Intel(R) Xeon(R) CPU E5-2620 v4 CPU. All reading and writing of data is done on temporary file storage. We use 20 threads to compress input images. Table 1 list two factors affecting efficiency: the evaluation time, which consists of both compression time for dataset and inference time of neural network, and number of trials required to generate first 10 good points. Good points are defined as points with accuracy larger than -0.001 of fitness

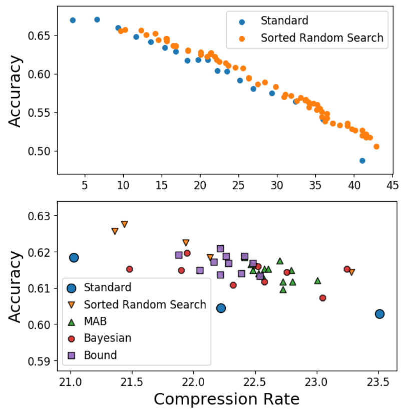

Figure 8. Pareto Performance Validated on Part of MatchedFrequency in ImageNetV2

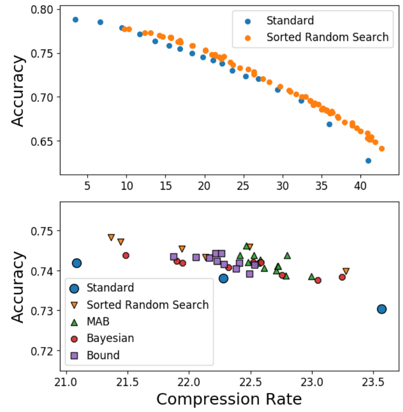

Figure 9. Pareto Performance Validated on Top Images in ImageNetV2

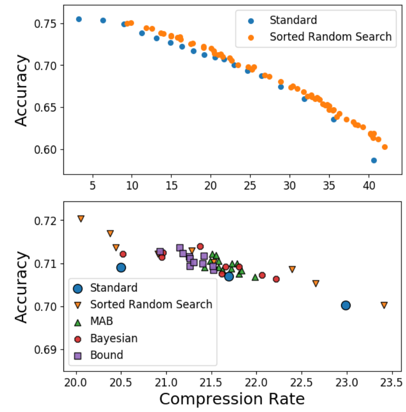

Figure 10. Pareto Performance Validated on ImageNet Validation Set

Method	Decision Time	Evaluation Time	# Trials for 10 Good Points
Random Search	$\approx$ 0.7ms	$\approx$ 180s	Inf
Sorted Random Search	$\approx$ 1.2ms	$\approx$ 180s	412
Bounded Random Search	$\approx$ 12ms	$\approx$ 180s	617
Bayesian Optimization w/ Local Grid Search	$\approx$ 60s	$\approx$ 180s	253
MAB	$\approx$ 5ms	$\approx$ 180s	150

Table 1. Efficiency among Algorithms

function $fitness(CR)$. Though the table, we can learn given the same compression rate as pareto of sorted random search, the trials the algorithm takes to find 10 points at least better than 0.1% of sorted random pareto accuracy. Bounded random search though has higher average than sorted random search, which is efficient in finding points approaching optimum, because it has structure information. MAB and Bayesian optimization has similar pareto optimal points but actually Bayesian has overhead in decision time since we applied local grid search. MAB also requires fewest trials to generate good points. An advantage of Bayesian, however, is it can easily retrieve past data by taking prior, and it should have increasing learning efficiency in few trials.

5.4 Generalization

In previous sections, we have identified good quantization table points based on a small portion of MatchedFrequency dataset of ImageNetV2. It is important to investigate if this quantization table perform consistent among other datasets.

Fig. 8 shows the performance of pareto points we find on 500 different classes with 5 images each of MatchedFrequency dataset from our training dataset. All quantization tables we take as pareto points outperform standard ones by 1% to 2% in accuracy when compression rate is the same. However, sorted random searched curve are not as smooth as our training curves. For bounded searching methods, Bayesian optimzation and MAB no longer obviously outperform sorted random and bounded random search.

Fig. 9 shows the performance of pareto points on the complete TopImage dataset in ImageNetV2. The quantization tables we find all outperform standard ones, but the margin compared to the result in Fig. 3 and Fig. 7 is reduced. The accuracy drops to 0.5% to 1% when compression rate is the same in which pareto points found by Bayesian optimization drops most. This shows though effective, our methods might be too restricted to training data.

We validate the quantization tables on ImageNet, with randomly chosen 10 images of each classes in the validation set to test the performance of quantization tables on other datasets. Fig. 10 shows accuracy gap between our quantization tables and standard ones is as close as 0.1% to 0.3% given the same compression rate. This might because re-compressing images that are already JPEG-compressed limits the exploration space for quantization table design, or the small ImageNetV2 MatchedFrequency dataset fail to generalize to the ImageNet dataset well.

5.5 Neural Network Retraining

Method	Compression Rate	Top-1 Accuracy
Standard (quality=50)	21.69	0.3784
Sorted Random Search	21.52	**0.3890**
	22.65	0.3843
Bounded Random Search	21.26	0.3776
	21.40	0.3887
Bayesian Optimization w/ Local Grid Search	21.37	0.3866
	21.61	0.3808
MAB	21.55	0.3829
	21.42	0.3848

Table 2. ImageNet retraining results for different algorithms

In all previous settings, we only train and test our quantization table on pretrained neural networks. We also wonder if retraining and testing on data compressed by the same quantization table would bring more information to the model and result in gain in accuracy. We pick the standard JPEG table of quality 50, as well as 2 pareto points each from aforementioned methods with closest compression rate to the standard one. For each of the selected tables, we compress the dataset with 100 images per classes from ImageNet, and train ResNet-50 with the following parameters: epochs of 10, batch size of 64, learning rate of 0.0002 for stochastic gradient descent optimizer.

Table 2 shows that under similar compression rate, most quantization tables by our algorithm outperform the standard JPEG table on top-1 accuracy by 0.5% to 1%. This is a preliminary experiment with small training set and few epochs, but it indicates a huge potential exploration space for neural network retraining.

6 Conclusion and Future works

Our research shows the potential in accuracy and compression rate improvement by redesigning JPEG quantization table, a topic that lacks attention of researchers in both JPEG and neural network communities. All our proposed methods effectively obtain better quantization tables compared to the standard one with an accuracy gain of 1% to 2% when compression rate fixed and a compression rate increase of 20% to 200% given the same accuracy. At training, MAB stands out with its high efficiency and accuracy. After cross-validation, all pareto points still outperform standard ones.But the gap between the standard JPEG and our proposed methods is decreased and no method demonstrated a dominance in performance advantage. The neural network retraining result shows the the potential impact of quantization table on neural network training and leave the space for further exploration.

References

Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., and Yu, D. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545, Oct 2014. ISSN 2329-9304. doi: 10.1109/TASLP.2014.2339736.

Ansel, J., Kamil, S., Veeramachaneni, K., Ragan-Kelley, J., Bosboom, J., O'Reilly, U.-M., and Amarasinghe, S. Opentuner: An extensible framework for program autotuning. In *International Conference on Parallel Architectures and Compilation Techniques*, Edmonton, Canada, August 2014. URL http://groups.csail.mit.edu/commit/papers/2014/ansel-pact14-opentuner.pdf.

Balasubramanian, V. K. and Manavalan, K. Knowledge-based genetic algorithm approach to quantization table generation for the jpeg baseline algorithm. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24 (3):1615–1635, 2016.

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. End to end learning for self-driving cars, 2016.

Collobert, R. and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 160–167, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390177. URL http://doi.acm.org/10.1145/1390156.1390177.

Costa, L. F. and Veiga, A. C. P. Identification of the best quantization table using genetic algorithms. In *PACRIM. 2005 IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2005.*, pp. 570–573. IEEE, 2005.

Cutello, V., Narzisi, G., and Nicosia, G. A multi-objective evolutionary approach to the protein structure prediction problem. *Journal of The Royal Society Interface*, 3(6): 139–151, 2005.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Li, F.-F. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Dodge, S. and Karam, L. Understanding how image quality affects deep neural networks, 2016.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Fung, H. T. and Parker, K. J. Design of image-adaptive quantization tables for jpeg. *Journal of Electronic Imaging*, 4 (2):144–151, 1995.

Hirschberg, J. and Manning, C. D. Advances in natural language processing. *Science*, 349(6245):261–266, 2015. ISSN 0036-8075. doi: 10.1126/science.aaa8685. URL https://science.sciencemag.org/content/349/6245/261.

Jiang, Y. and Pattichis, M. S. Jpeg image compression using quantization table optimization based on perceptual image quality assessment. In *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 225–229. IEEE, 2011.

Kaur, B., Kaur, A., and Singh, J. Steganographic approach for hiding image in dct domain. *International Journal of Advances in Engineering & Technology*, 1(3):72, 2011.

Kumar, B. V. and Karpagam, G. Knowledge-based differential evolution approach to quantisation table generation for the jpeg baseline algorithm. *International Journal of Advanced Intelligence Paradigms*, 8(1):20–41, 2016.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft coco: Common objects in context, 2014.

Liu, Z., Liu, T., Wen, W., Jiang, L., Xu, J., Wang, Y., and Quan, G. Deepn-jpeg: A deep neural network favorable jpeg-based image compression framework. In *Proceedings of the 55th Annual Design Automation Conference*, pp. 18. ACM, 2018.

LuaDist. Luadist/libjpeg, Nov 2015. URL https://github.com/LuaDist/libjpeg.

Ma, H. and Zhang, Q. Research on cultural-based multi-objective particle swarm optimization in image compression quality assessment. *Optik-International Journal for Light and Electron Optics*, 124(10):957–961, 2013.

Naresh, S., Kumar, B. V., and Karpagam, G. A literature review on quantization table design for the jpeg baseline algorithm. *International Journal of Engineering And Computer Science*, 4(10):14686–14691, 2015.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Ramchandran, K. and Vetterli, M. Rate-distortion optimal fast thresholding with complete jpeg/mpeg decoder compatibility. *IEEE Transactions on image processing*, 3(5): 700–704, 1994.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.

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.

Szegedy, C., Toshev, A., and Erhan, D. Deep neural networks for object detection. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 2553–2561. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf.

Tuba, M. and Bacanin, N. Jpeg quantization tables selection by the firefly algorithm. In *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 153–158. IEEE, 2014.

Tušar, T. and Filipič, B. Differential evolution versus genetic algorithms in multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 257–271. Springer, 2007.

Wallace, G. K. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

Wang, C.-Y., Lee, S.-M., and Chang, L.-W. Designing jpeg quantization tables based on human visual system. *Signal Processing: Image Communication*, 16(5):501–506, 2001.

Watson, A. B. Dctune: A technique for visual optimization of dct quantization matrices for individual images. In *Sid International Symposium Digest of Technical Papers*, volume 24, pp. 946–946. Society for information display, 1993.

Westen, S. J., Lagendijk, R. L., and Biemond, J. Optimization of jpeg color image coding using a human visual system model. In *Human Vision and Electronic Imaging*, volume 2657, pp. 370–381. International Society for Optics and Photonics, 1996.

Wikipedia contributors. Flickr — Wikipedia, the free encyclopedia, 2019. URL https://en.wikipedia.org/w/index.php?title=Flickr&oldid=929460302. [Online; accessed 8-December-2019].

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb 2009. ISSN 1939-3539. doi: 10.1109/TPAMI.2008.79.

Wu, S.-W. and Gersho, A. Rate-constrained picture-adaptive quantization for jpeg baseline coders. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pp. 389–392. IEEE, 1993.

Wu, Y.-G. Ga-based dct quantisation table design procedure for medical images. *IEE Proceedings-Vision, Image and Signal Processing*, 151(5):353–359, 2004.