
OPTIMIZING JPEG QUANTIZATION FOR CLASSIFICATION NETWORKS

Zhijing Li¹ Christopher De Sa¹ Adrian Sampson¹

ABSTRACT

Deep learning for computer vision depends on lossy image compression: it reduces the storage required for training and test data and lowers transfer costs in deployment. Mainstream datasets and imaging pipelines all rely on standard JPEG compression. In JPEG, the degree of quantization of frequency coefficients controls the lossiness: an 8×8 *quantization table* (*Q-table*) decides both the quality of the encoded image and the compression ratio. While a long history of work has sought better Q-tables, existing work either seeks to minimize image distortion or to optimize for models of the human visual system. This work asks whether JPEG Q-tables exist that are “better” for specific vision networks and can offer better quality–size trade-offs than ones designed for human perception or minimal distortion.

We reconstruct an ImageNet test set with higher resolution to explore the effect of JPEG compression under novel Q-tables. We attempt several approaches to tune a Q-table for a vision task. We find that a simple *sorted random* sampling method can exceed the performance of the standard JPEG Q-table. We also use hyper-parameter tuning techniques including bounded random search, Bayesian optimization, and composite heuristic optimization methods. The new Q-tables we obtained can improve the compression rate by 10% to 200% when the accuracy is fixed, or improve accuracy up to 2% at the same compression rate.

1 INTRODUCTION

Deep neural networks for vision have extreme storage requirements. For example, ImageNet is 150 GB (Deng et al., 2009) and the Open Image Dataset requires 500 GB (Kuznetsova et al., 2018). If a model can achieve equivalent performance with images at a higher compression ratio, it can be cheaper to acquire, transmit, and store the necessary datasets.

Nearly all DNN datasets use the ubiquitous JPEG compression standard, including ImageNet (Deng et al., 2009), Pascal VOC (Everingham et al.), COCO (Lin et al., 2014), etc. Because JPEG is a loss compression format, DNNs cope with some quality distortion in these images (Dodge & Karam, 2016). However, the JPEG standard is tuned for the human visual system (HVS) and preserves the components that are most relevant to human observers, rather than to computer vision algorithms. This work asks whether machine learning pipelines would perform better if JPEG were instead tuned specifically for a given DNN.

Fig. 1 shows an overview of the standard JPEG compression

algorithm. Each spatial component (i.e., each YCbCr channel) is first partitioned into 8×8 non-overlapping blocks. Then the spatial components are transformed to frequency components with the 2D discrete cosine transform (DCT). Next, the algorithm quantizes the frequency components using a quantization table or *Q-table*. The Q-table’s scale is determined by the *quality factor* ranging from 1 to 100 (Lu-aDist, 2015). For instance, a quality factor of 100 scales Q-table coefficients to 0 and the frequency components are rounded to integers. Then, the quantized coefficients are serialized using a *zig-zag* ordering and losslessly compressed with entropy coding and Huffman coding.

Quantization is the lossy part of the algorithm. The Q-table decides how many bits to use for each frequency bucket, potentially playing a critical role for DNNs that use JPEG-compressed images. However, existing research on designing Q-tables mostly prioritizes human-perceptible distortion as the compression target (Liu et al., 2018), which may compress and quantize features important to DNNs (Wright et al., 2009). Prior work instead targets DNN accuracy (Liu et al., 2018), but it only considers *recompressing* existing JPEG-compressed datasets rather than changing the way data is compressed in the first place.

In this work, we use a high-resolution dataset to simulate the effect of compressing raw pixels with novel, task-specific quantization. We use a range of optimization methods to tune a customized Q-table for a specific DNN that max-

¹Cornell University. Correspondence to: Zhijing Li <zl679@cornell.edu>.

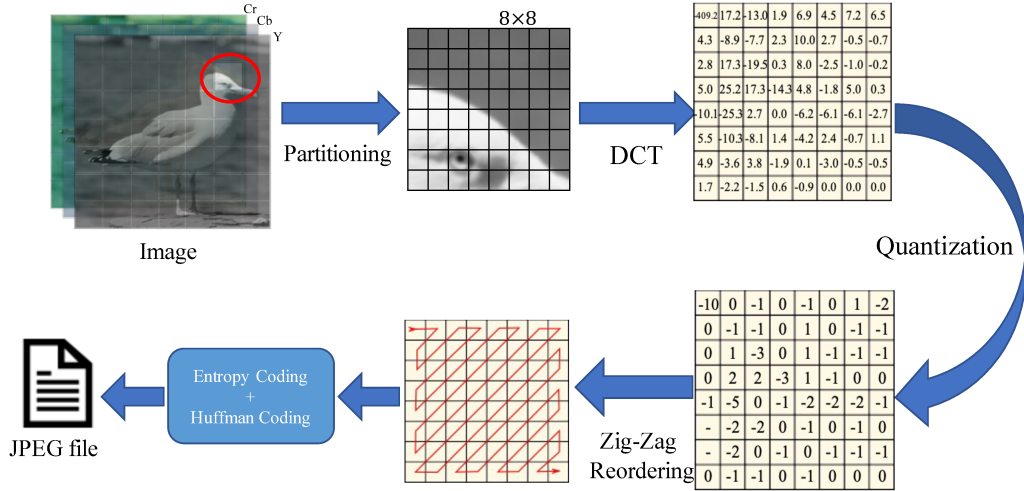


Figure 1. Overview of the JPEG compression algorithm.

imizes both vision task accuracy and compression ratio. Starting with a simple baseline method, *sorted random search*, we find Q-tables that improve ResNet50 top-1 accuracy by 1–2% at the same compression rate as standard JPEG or improve compression rate by 20–200% at the same accuracy. We then use a range of hyper-parameter tuning methods, including Bayesian optimization and a composite heuristic autotuner, to improve on these Q-tables within bounds derived from the best results from the baseline method. We find that these autotuning approaches yield small improvements over the simple approach, but after cross validation, the methods perform similarly. Surprisingly, our simple sorted random search approach performs as well as the more advanced tuning approaches, and the advantages over standard JPEG are statistically significant.

The contributions of this paper are:

- We construct a high-resolution dataset for experimenting with novel compression techniques based on ImageNetV2 (Recht et al., 2019).
- We find that there is an opportunity to save space for the same accuracy by customizing the JPEG quantization table.
- In particular, we recommend using a sorted random search procedure, which quickly produces tables that meet and exceed the standard JPEG quantization tables.
- Despite extensive experimentation with other optimization procedures, we did not find a method that obviously outperforms sorted random search.

2 RELATED WORK AND MOTIVATION

2.1 Q-Table Optimization

Because JPEG is one of the most widely used image compression techniques, JPEG Q-table optimization has been

an enduring issue. Existing approaches to design JPEG Q-tables aim to maximize image quality in one of two ways: by minimizing a simple distortion metric such as PSNR (Wu & Gersho, 1993; Fung & Parker, 1995; Ramchandran & Vetterli, 1994) or by using a model of the human visual system (HVS) (Watson, 1993; Wang et al., 2001; Westen et al., 1996; Jiang & Pattichis, 2011). The HVS approach aims at optimizing visual image quality, where high-frequency components are viewed as less important than low-frequency components. Standard JPEG is designed with the same goal (Wallace, 1992). This work’s hypothesis is that the best Q-tables for DNN vision may be different than ones that emphasize visual quality because of different sensitivity to frequency components (Liu et al., 2018).

2.2 DeepN-JPEG

There is some work that focuses on the optimization of Q-tables for DNNs. DeepN-JPEG (Liu et al., 2018) exploits the difference between the HVS and DNNs and designs a Q-table according to the frequency bands. The Q-table search space is limited by design: the frequency bands share a small number of parameters. We explore a larger space using hyper-parameter tuning. Also, DeepN-JPEG both tunes and tests on ImageNet, without cross validation on another dataset. ImageNet is a low-resolution, compressed dataset, and that work *recompresses* it. We instead focus on simulating compression of raw pixel data using a high-resolution rebuild from ImageNetV2 (Recht et al., 2019) and validate our result on different datasets to check robustness.

3 VISION TASK

For the purpose of tuning Q-tables for DNNs, we need to choose a particular vision task. We use a classification network, ResNet50 (He et al., 2016), and reconstruct a

dataset to simulate raw pixel data.

3.1 Constructing a Dataset

ImageNet is already downsized and lossily compressed. For instance, the 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 have open-sourced the code to generate the dataset as well as the IDs that let us re-download the original, large images from Flickr (Fli). The large images are also JPEG-compressed, but they have an average resolution of 1933×1592 pixels. We then downsize them to a smaller size compatible with ImageNet. The resizing removes JPEG compression artifacts and simulates uncompressed images at those sizes.

ImageNetV2 contains three test sets representing different sampling strategies: MatchedFrequency, TopImages, and Threshold0.7. Each test set has 1000 classes with 10 images per class. Specifically, we use the MatchedFrequency as our training dataset, which is sampled to match the MTurk selection frequency distribution of the original ImageNet validation set for each class.

3.2 Tuning Methodology

Hyper-parameter tuning for all kinds of DNNs is unrealistic. We focus on one particular DNN: ResNet50 (He et al., 2016) as implemented in PyTorch (Paszke et al., 2017). We use 500 classes with 5 images each from ImageNetV2’s MatchedFrequency dataset to speed up compression. The optimization targets are top-1 accuracy and the compression rate, calculated as the ratio of the size of raw bitmap data to compressed JPEG images.

4 SORTED RANDOM SEARCH

We start with a simple method that randomly samples Q-tables. The idea is to get a baseline to see how hard it is to find, among the space of all Q-tables, new ones that mimic the performance of the standard JPEG tables—or that outperform standard JPEG for a specific DNN for a specific vision task.

4.1 Method

The search space for a uniform random search is $256^{64} \approx 1.34 \cdot 10^{154}$. We need to decrease our search space. Lower frequencies probably matter more than higher frequencies. That is why the upper left of the standard table are large and the lower right are small, as shown in Fig. 1. A simple sorting strategy can restrict sampling to tables that preserve

this ordering. We generate 64 random numbers in the range $[s, e]$, where $s, e \in \mathbb{Z}$ and $1 \leq s < e \leq 255$, sort them, and build an 8×8 table using the zig-zag ordering. We refer to this Q-table sampling strategy as *sorted random search*.

4.2 Performance

Fig. 2 compares 4000 Q-tables sampled using sorted random search, 1000 from uniform random search, and the standard Q-tables with quality factor from 10 to 100 at an interval of 5 in terms of compression rate and accuracy. We also plot the Pareto frontier for sorted random search. With uniform random search, we cannot find any Q-table that outperforms the standard JPEG Q-table. Sorted random search finds Pareto-optimal points that outperform standard JPEG by 1% to 2% in top-1 accuracy given the same compression rate when the quality factor of standard JPEG is in the range $[15, 90]$. Given the same accuracy, sorted random search can find Q-tables with compression rates 10% to 200% higher than standard JPEG depending on the baseline compression rate. We further take a close look at points with closest compression rate and accuracy to the quality-50 Q-table of standard JPEG, where the standard table is scaled at 100%. The improvements of accuracy and compression rate are 1.5% and 12.9%.

Fig. 3 shows the distribution of sorted random points with compression rate between 22 to 22.2. Among 52 points, there are 48 points that perform at least as well as the standard JPEG Q-table, of which 93.75% perform better. Taking into account the strong randomness of sorted random search, the result indicates sorted random search can easily find many Q-tables that outperform standard JPEG.

To demonstrate the difference between the PSNR and DNN optimization goals, Fig. 4 plots the PSNR of sorted random search points and we highlight the Pareto frontier in terms of compression rate and accuracy. Although PSNR is closely related to DNN accuracy, it cannot be taken as the sole indicator of high accuracy.

5 BOUNDED SEARCH

The random search method from the previous section samples from a tremendously large search space but uses an ordering constraint to make the search tractable. In this section, we hypothesize that, if we limit our search space of Q-table components to a smaller range, we can eliminate ordering constraint and find better tables using more sophisticated optimization methods.

To limit the search space, we need to focus on a particular compression rate range. We focus on a regime close to standard JPEG at quality 50, where the compression rate is approximately 22. Let P be the set of Q-tables on the Pareto frontier and $P' \subseteq P$ be the set of Q-tables with com-

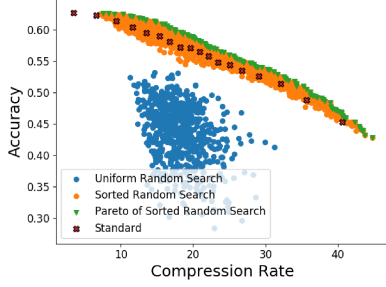


Figure 2. Sorted random search and random search performances compared to standard JPEG.

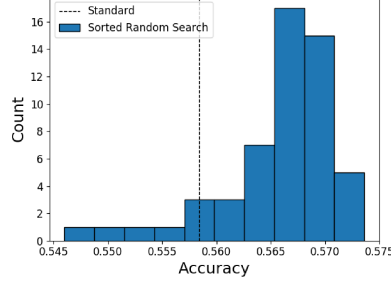


Figure 3. Sorted random search distribution with compression rate range (22, 22.2).

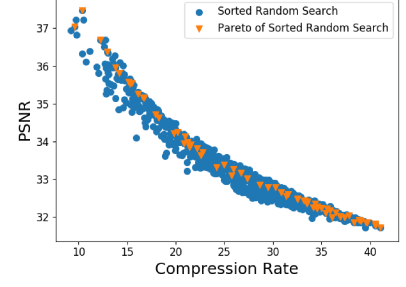


Figure 4. PSNR of sorted random search with Pareto-optimal points according to DNN accuracy.

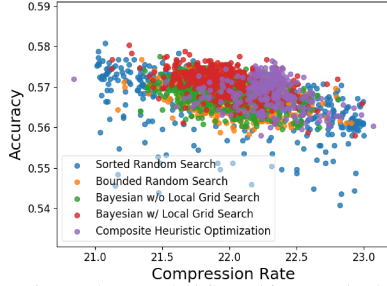


Figure 5. Bounded Searching Methods

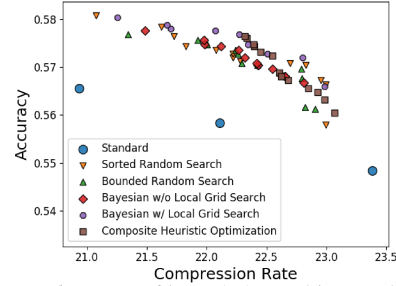


Figure 6. Pareto of bounded searching methods.

pression rate ranges in $[21, 23]$. Let P'' be the set of Q-tables in P' and their transpose. Let $M_{P''}$ be the $n \times 8 \times 8$ array that results from concatenating the Q-tables in P'' . We can get a symmetric upper bound matrix by taking maximum along first dimension of $M_{P''}$. Therefore, the 8×8 boundary matrices are set to:

$$M_{\text{Lower Bound}}[j, k] = \min_i (M_{P''}[i, j, k]) - 0.5\sigma_i(M_{P''}[i, j, k])$$

$$M_{\text{Upper Bound}}[j, k] = \max_i (M_{P''}[i, j, k]) + 0.5\sigma_i(M_{P''}[i, j, k])$$

where $i \in [0, n-1]$, $j \in [0, 7]$, $k \in [0, 7]$ and σ_i takes the standard deviation along first dimension of a matrix. We can sample Q-tables with the two boundary matrices to limit the exploration space. We expect that Q-tables sampled within these boundaries will result in compression rates in the range $[21, 23]$.

5.1 Bounded Random Search

The most intuitive way as a baseline for bounded search is uniformly sampling Q-tables within the boundaries, which we called *bounded random search*. We plot Q-tables sampled using bounded random search in Fig. 5 and corresponding Pareto frontiers in Fig. 6.

5.2 Bayesian Optimization

Bayesian optimization is widely used when evaluation of the objective function is expensive. It approaches the objective function with a surrogate model, which is cheap to evaluate.

To take the advantage of Bayesian optimization, there are two problems we need to address.

First, Bayesian optimization optimizes a single target objective function and our task involves two: compression rate CR and accuracy ACC . We therefore use a parabola $\text{fitness}(CR) = aCR^2 + bCR + c$ to fit the Pareto frontier formed by Q-tables obtained in sorted random search. Then we can set the target value $y = ACC - \text{fitness}(CR)$, standing for the difference between the accuracy gap between actual accuracy and best accuracy found by sorted random search.

Second, the search space is huge—as large as $1.34 \cdot 10^{154}$. We therefore set the sampling bound the same as the one in the previous section. Also, the low frequency (LF) and middle frequency (MF) components of DCT 8×8 arrays have larger absolute values (Kaur et al., 2011), and therefore are more susceptible to changes in quantization values. We define the *area of interest* as the low frequency and middle frequency bands as shown in Fig 10. We first generate 10^5 random Q-tables and keep the one with highest acquisition function value. Then we randomly choose 5 indices of that Q-table in the area of interest and perform a local grid search that exhaustively list possible Q-tables within the bound. The Q-table with highest acquisition function value is kept. The process is repeated 20 times. In our experiments, we try Bayesian optimization both with and without local random search for comparison and plot them in Figs. 5 and 6.

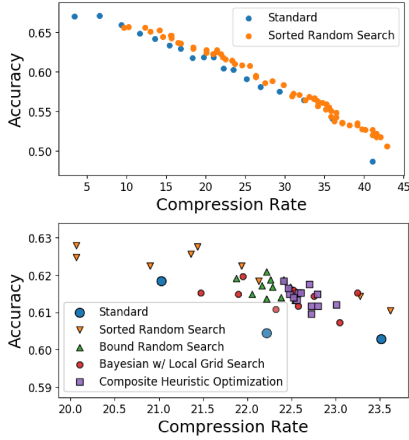


Figure 7. Pareto performance validated on MatchedFrequency (part) in ImageNetV2.

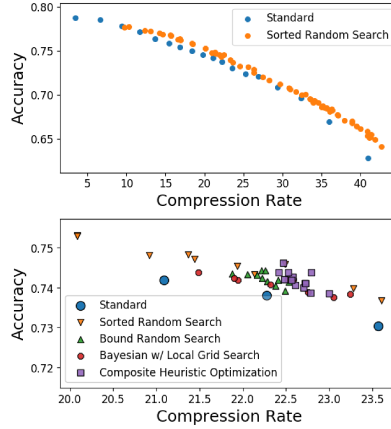


Figure 8. Pareto performance validated on TopImages in ImageNetV2.

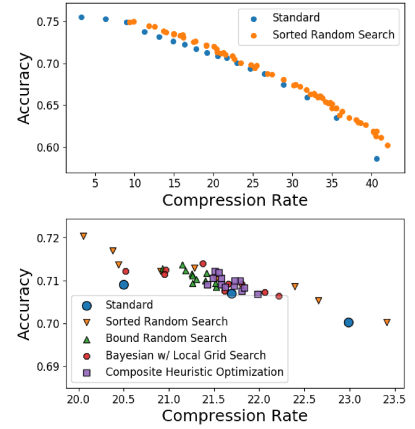


Figure 9. Pareto performance validated on ImageNet validation set.

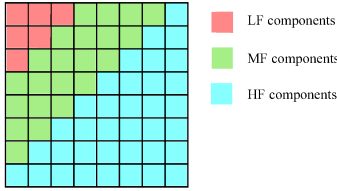


Figure 10. Frequency band partitions.

5.3 Composite Heuristic Optimization

Different search methods have different strengths. We next apply OpenTuner (Ansel et al., 2014), a generic auto-tuning framework that uses a *multi-armed bandit* (MAB) approach to select between several methods. OpenTuner maximizes an objective by picking a strategy among different optimization approaches (the bandit *arms*). We select the particle swarm optimization (PSO), simulated annealing (SA), differential evolution (DE), greedy mutation, and random Nelder-Mead heuristics as the bandit arms. All these heuristics are generic optimization algorithms and can be performed without much computation overhead.

Figs. 5 and 6 show that this method produces results that perform as successfully as Bayesian optimization.

5.4 Performance

Fig. 5 shows 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 composite heuristic optimization. We also plot the points from sorted random search in the range $[21, 23]$ as a baseline. Fig. 6 shows that the Pareto frontiers of all methods and standard JPEG. All proposed methods can find Q-tables that outperform standard JPEG. Simple techniques like bounded random search

can give a better distribution than sorted random search, but the Pareto frontier does not improve. Bayesian optimization without local grid search does not show a strong advantage over sorted and bounded random search mainly because the sampling space is too large to find a promising point to probe. Bayesian optimization with local grid search and composite heuristic optimization are the two methods with the highest means, smallest standard deviations, and the best Pareto frontiers. Given the same compression rate, the accuracy improvement can be up to 2.5% over standard JPEG. In particular, the point with the closest compression rate to quality-50 standard JPEG is found by Bayesian optimization, and the absolute accuracy improvement is 1.9%.

6 EVALUATION

This section examines accuracy under cross validation and the efficiency of each method.

6.1 Cross Validation

The experiments above find good Q-tables using part of ImageNetV2’s MatchedFrequency dataset. It is important to check whether these Q-tables perform consistently on other datasets. We cross-validate our Q-tables on the other 500 classes of the reconstructed MatchedFrequency dataset with 5 images each in Fig. 7, the reconstructed TopImage dataset of ImageNetV2 in Fig. 8, and the original ImageNet dataset with 10 images per class in Fig. 9.

The Pareto frontier of sorted random search becomes less smooth. When the compression rate is the same, the gap of accuracy improvement between our methods and standard JPEG decreases to at most 1.5% for the MatchedFrequency and TopImages datasets (Figs. 7 and 8) and 1% for ImageNet (Fig. 9) but never disappears. Bayesian optimization and MAB no longer significantly outperform sorted random and

Table 1. Significance of improvement over baseline, where * denotes mean differences that are statistically significant according to a Student t -test with $p < 10^{-5}$ and ** denotes $p < 10^{-11}$.

	MATCHED FREQUENCY	IMAGENET
SORTED RANDOM SEARCH	0.91%**	1.16%**
BOUNDED RANDOM SEARCH	0.72%*	0.66%*
BAYESIAN OPTIMIZATION W/ LOCAL GRID SEARCH	0.55%*	0.77%*
COMPOSITE HEURISTIC OPTIMIZATION	0.73%*	1.17%**

bounded random search. Overall, after cross validation, the complex algorithms do not show any advantage over our simple baseline algorithm, sorted random search.

6.2 Significance of Improvement

To check whether the improvement of our methods over standard JPEG is significant, we perform statistical analysis of accuracy improvement given the closest compression rate. We test one Q-table obtained by each method against standard JPEG with quality set to 50. To avoid unintentionally cheating, we choose the point with the closest compression rate that is *larger* than that for the quality-50 standard Q-table, i.e., we allow accuracy to be a little worse. For each Q-table, we randomly sample 100 datasets from the original ones with 700 classes and 4 images per class. Then we compare the difference in mean accuracy and check significance using Student’s t -test.

Table 1 shows the difference of mean top-1 accuracy between standard JPEG and our Q-table obtained by different methods, on the MatchedFrequency and ImageNet Validation dataset. All methods find Q-tables with statistically significant accuracy improvement.

6.3 Efficiency

Dataset compression is expensive. For instance, the time to compress our training dataset and measure accuracy is approximately 3 minutes. A good method should find optimal Q-tables in as few trials as possible and the decision time for each trial should be short.

Table 2 shows the results of profiling each method on a server with two Intel Xeon(R) E5-2620 v4 CPUs with 8 cores per socket and 2 threads per core. The table lists two factors affecting efficiency: the decision time for an algorithm to decide the next point to search, and the number of trials required to generate the first 10 good points. “Good” points are defined as those with $\text{Acc} - \text{fitness}(\text{CR}) > -0.001$. The decision time is given as the average over 100 trials, rounded to two significant digits. Among all methods,

Table 2. Efficiency of different methods.

METHOD	DECISION TIME	TRIALS FOR 10 GOOD POINTS
SORTED RANDOM SEARCH	1.2MS	412
BOUNDED RANDOM SEARCH	12.MS	617
BAYESIAN OPTIMIZATION W/ LOCAL GRID SEARCH	60.S	253
COMPOSITE HEURISTIC OPTIMIZATION	5.1MS	150

composite heuristic optimization requires the fewest trials to generate good points, while bounded random search takes the most trials to find good points. The decision time for all except Bayesian optimization is negligible, as the latter requires an expensive local grid search.

7 CONCLUSION AND FUTURE WORK

Our research shows the potential for accuracy and compression rate improvement by redesigning the JPEG Q-table, a topic that lacks attention from researchers in both of the image compression and DNN communities. All our proposed methods obtain better Q-tables compared to the standard one with an accuracy gain of 1% to 2% when the compression rate is fixed and a compression rate increase of 10% to 200% given the same accuracy. Although a composite heuristic optimizer stands out in efficiency and accuracy, we recommend a simple sampling technique, *sorted random search*, which can find good quantization tables at a wide range of compression–accuracy trade-offs. The resulting improved Q-tables outperform standard JPEG for our vision task in cross validation.

In this work, we consider the DNN fixed and only tune compression. But the network was trained on data using standard JPEG and therefore might perform worse on images using our new Q-tables. Fine-tuning retraining might help adjust the networks to match the new image characteristics, further improving the accuracy with nonstandard quantization. We have done a preliminary retraining experiment, which shows a positive result. Avenues for future work include further work on DNN retraining and applying our techniques to more vision applications such as object detection, semantic segmentation, and other tasks.

ACKNOWLEDGMENTS

This research was supported by gifts from Google and Nvidia. We thank our colleagues Mark Buckler, Philip Bedoukian, and Jenna Choi, who initiated this research direction. We also thank Zhiyong Hao and Danna Ma for comments that improved the manuscript. This work was supported by NSF award #1909073.

REFERENCES

- About Flickr. <https://www.flickr.com/about>. Accessed: 2020-01-12.
- 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>.
- 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. In *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, 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.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- 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.
- Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., Duerig, T., et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.
- 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>.
- 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.
- 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.
- 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.