

Deep Layers Beware: Unraveling the Surprising Benefits of JPEG Compression for Image Classification Pre-processing

Guruprasad Nayak, Gerald Friedland
Amazon AI
nayaguru@amazon.com, gfriedla@amazon.com

Abstract—In this paper, we explore the intriguing effects of JPEG compression as a pre-processing technique for image classification tasks. Building upon the findings of a previous study by Friedland et al., which demonstrated that substantial JPEG compression does not significantly degrade classification accuracy, we investigate the potential benefits and limitations of this approach when applied to various classifiers, such as AutoGluon-multimodal and EfficientNet. Our experiments not only confirm the original results but also reveal notable I/O benefits, with compressed images occupying as little as 14 % of the original dataset size while maintaining comparable accuracy.

Despite these promising findings, we also document several investigations that did not yield beneficial outcomes. We found no evidence to suggest that JPEG compression leads to faster model convergence or allows smaller models to achieve the same accuracy. Additionally, our experiments showed that tabular classifiers could not match the performance of deep neural networks when trained on JPEG-compressed input, and that JPEG compression does not make classifiers more resilient to noise in input images.

Together, our results provide a comprehensive evaluation of JPEG compression as a pre-processing technique for image classification. While the approach offers undeniable benefits in terms of data storage and accuracy preservation, it does not appear to yield advantages in terms of model convergence, model size, or robustness to noise. This study contributes valuable insights for researchers and practitioners working in multimedia signal processing and image recognition, paving the way for further exploration and optimization of multimedia compression techniques.

I. INTRODUCTION

In recent years, deep learning-based multimedia recognition techniques have revolutionized the field of multimedia computing, enabling breakthroughs in a wide range of applications, such as autonomous vehicles [1], medical imaging [2], and surveillance systems [3]. As the demand for large-scale multimedia analysis continues to grow, efficient processing and storage of multimedia data have become critical challenges. One way to address these challenges is through multimedia compression, which reduces the size of image files and, consequently, the computational resources required to process them.

Widely-used lossy multimedia compression formats, such as JPEG [4] for images and MPEG [5] for videos, have been the subject of several studies investigating their impact on the performance of deep learning algorithms for multimedia recognition tasks [6], [7], [8]. Surprisingly, recent research [7]

has shown that even substantial JPEG compression does not incur significant drops in classification accuracy until very high compression levels are reached. This observation raises intriguing questions about the potential benefits of JPEG compression as a pre-processing technique for image classification models.

In this paper, we delve deeper into the implications of using JPEG compression for pre-processing images before feeding them to various classifiers, as shown in figure 1, such as AutoGluon-multimodal and EfficientNet. Our comprehensive analysis confirms the previous findings [7], demonstrating that substantial JPEG compression can be applied to input images without significantly affecting classification accuracy. Furthermore, we highlight the I/O benefits of this approach, as compressed images can occupy as little as 14 % of the original dataset size while maintaining comparable accuracy.

Despite these advantages, our study also investigates several aspects of JPEG compression that did not yield beneficial outcomes. We found that JPEG compression does not lead to faster model convergence, enable smaller models to achieve the same accuracy, enhance the performance of tabular classifiers, or make classifiers more resilient to noise in input images.

The research presented in this article is closely related to several studies that have investigated the impact of image compression including JPEG on deep learning-based image recognition [9], [10], [11], [12], [7]. Building on these previous studies, our work contributes valuable insights into the potential benefits and limitations of using JPEG compression as a pre-processing technique for image classification tasks. By providing a comprehensive analysis of the impact of JPEG compression on deep learning-based image recognition, we aim to support researchers and practitioners in designing more efficient and robust systems that can better cope with the challenges posed by large-scale multimedia analysis. Furthermore, our findings open up new avenues for further exploration and optimization of multimedia compression techniques in the context of multimedia signal processing and image recognition.

II. EXPERIMENTS - POSITIVE FINDINGS

In this section, we present experiments to study the applicability of JPEG compression to learning image classification

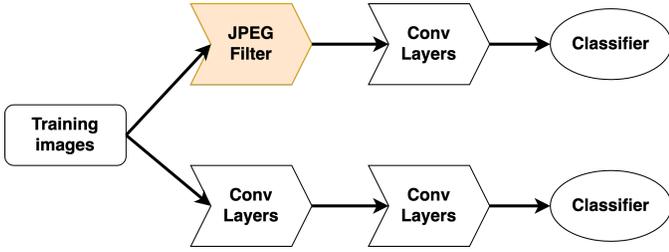


Fig. 1: Workflow of noise reduction performed by either JPEG compression or alternatively by more convolutional layers.

models. Table I shows the data sets we consider to test our hypotheses.

TABLE I: Data sets used for experiments

Dataset	#Classes	#Images			
		Total	Train	Test	Validation
CIFAR100	100	60,000	36,000	12,000	12,000
Caltech256	257	30,592	18,356	6,119	6,117
OxfordIIITPets	37	7,400	4,440	1,480	1,480
SUN397	397	108,754	65,251	21,751	21,752
DTD	47	5,640	3,384	1,128	1,128
CIFAR10	10	60,000	36,000	12,000	12,000

A. Effect of JPEG compression on classification accuracy

In this experiment, we document the effect of compressing the input images on the accuracy of the classification model trained subsequently. For every data set, we do the following

- 1) We first construct different compressed versions of it, each corresponding to a different compression ratio, ranging from no compression (ratio = 1) to highest compression (ratio = 0.1). We consider 10 compression ratios in all.
- 2) We then train a deep neural network model on the data set by fine-tuning a pre-trained model from a model zoo for a fixed number of epochs. Through this process, we get 10 trained models, one for each level of compression.
- 3) Each model is evaluated on the test set and its accuracy is documented.

The performance of the model trained on different levels of compression in the input data can be plotted as a curve with accuracy on the vertical axis and compression ratio on the horizontal axis. Figure 2 shows the performance x compression curves for different data sets. Here, we use Autogluon-Multimodal [13] as the classifier. We observed similar results with other classifiers including those from the EfficientNet [14] family of models.

We observe that, every data set has a certain amount of compression, that it can tolerate before the amount of relevant information in the image for the classification task is reduced significantly enough that we start seeing a drop in

the performance on the test set.

We observe that for all data sets, except for CIFAR10 and CIFAR100, we see negligible drop in performance until around the compression ratio 0.3. In other words, for these data sets, we can reduce 70% of the information content in the image and still maintain the same classification performance.

For data sets CIFAR10 and CIFAR100, we hypothesize that the because the images are already very small (32*32 size images of size 1-2 KBs), they do not have any noise for the JPEG algorithm to compress. Therefore, applying JPEG would suppress the useful signal in the image relevant for the classification task. Thus, we see that even a small amount of compression drops the performance of the classifier significantly.

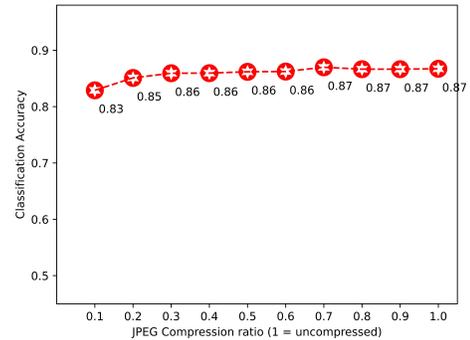


Fig. 2: Variation in classification performance with different levels of compression. For most data sets, we observe negligible drop in accuracy until JPEG compression ratio 0.3. For CIFAR10, CIFAR100 data sets, since the original images are very small to begin with, applying JPEG would suppress the useful signal in the image relevant for the classification task. Thus, we see that even a small amount of compression drops the performance of the classifier significantly for them. (Representative results only included for Caltech256 data set)

B. Relative risk between JPEG-compressing input versus increasing model size

In this experiment, we document the performance of classification models of different sizes with compressed and uncompressed input images. To remove any effects on the performance due the model architecture, we consider models of different sizes from the same model family (EfficientNet). For each data set, we do the following -

- 1) Create a compressed version of the data set with 0.3 compression ratio. We choose 0.3 compression ratio since that is when the performance starts to drop for most of our data sets (as seen in section II-A)
- 2) Train 5 models of EfficientNet family (B0, B1, B2, B3, B4) on the compressed and uncompressed data set, with model size varying from 5.3M to 19M parameters. We take the pre-trained weights for each model from a

model zoo (such as timm) and fine-tune it on the data set with a budget of 10 epochs.

- 3) The accuracy of each model is documented on the test set, for both the compressed input case and the uncompressed input case.

We observe that except for the CIFAR10 data set, the accuracy of the smallest model (EfficientNet-B0) is greater than the accuracy of the largest model (EfficientNet-B4), when trained for the same number of epochs. This benefit is summarized in table II. This shows that drop in performance with JPEG-compressing input images is significantly lower in comparison to the drop in performance observed when larger deep neural network models are trained, keeping the same epoch-budget during training.

Given that increasing the number of parameters is a standard way to boost classification performance, it can be seen that JPEG-compressing input images for classification is relatively low-risk, from an accuracy standpoint.

C. I/O benefits of JPEG-compressing input for classification

One of the key benefits of JPEG-compressing the input images for classification is the reduction in the time spent on I/O operations with the images. Since the I/O operation time is very system-dependent, we capture this benefit by quantifying the drop in storage space on disk required by the original data set and its compressed version. In this experiment, we do the following for each data set -

- 1) We create a compressed version of the data set with compression ratio 0.3. We choose 0.3 compression ratio since that is when the performance starts to drop for most of our data sets (as seen in section II-A)
- 2) We train a EfficientNet-B0 model on the compressed and uncompressed version of the data set. Training is performed by taking pre-trained model weights from a model zoo and fine-tuning for 10 epochs.
- 3) Performance of the trained models on the test set and storage space taken by the data sets on disk are reported.

Table III shows the results of this experiment. We can see that, we get anywhere between 65% to 86% reduction in storage space required for the input. In most data sets (except for CIFAR), this storage space reduction comes with negligible drop in performance.

III. EXPERIMENTS - NEGATIVE FINDINGS

JPEG compression is designed to reduce the effect of less important information from the input images. This is done by discretizing the data and rounding off the higher frequency components of the Discrete Cosine Transform (DCT) of the data. Thus, we wanted to test the following hypotheses with JPEG-compressed input -

- 1) How well does JPEG-compression handle noise in the input data?
- 2) Does reducing irrelevant information in the input data via compression help train classification models faster?

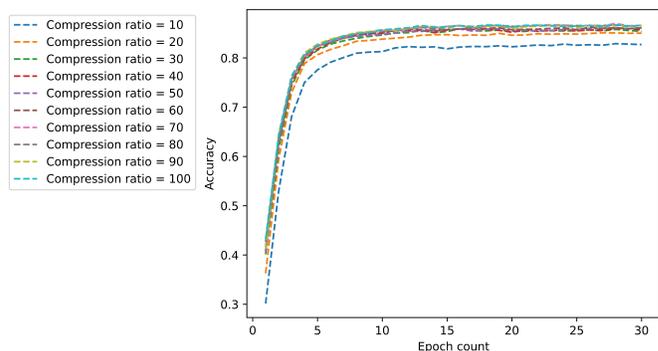


Fig. 3: Accuracy per epoch over different levels of compression. Testing if the accuracy tapers with less epochs with compressed data. Result on Caltech256 data set. Similar results were observed for other data sets.

A. Handling noise with compression

1) *Robustness to data noise: adding perturbations in input images:* In this experiment, we added multiplicative Gaussian noise to input images and documented the performance of AutoGluon system (for image classification) with and without JPEG-preprocessing. The hypothesis of this experiment was the following - adding noise to the training data will reduce the accuracy of the trained classifier. However, if we JPEG preprocess the noisy training data, it should help recover the lost performance. We thought this could work because JPEG removes noise from the input. However, the experiment did not support this hypothesis.

How is the noisy image created?

The input image is first normalized to have values between 0 and 1. Then, the noisy image is created like so,

$$noisy_image = image + image * gaussian_noise$$

where *gaussian_noise* is a random number drawn from a standard normal distribution ($mean = 0, std = 1$). Here is an example of a clean image versus a noisy version of the same image. Table IV shows the results of training classifier on noisy input and JPEG-compressed noisy input. We can see that JPEG compressing the noisy input does not help salvage the drop in accuracy resulting from the noise in the input.

B. Convergence of training deep neural networks with compressed input

In this experiment, we observed the convergence of models across different levels of compression of input data. Figure 3 shows the results of this experiment. We see that the accuracy curve for uncompressed input (compression ratio = 100) plateaus at the same rate as the curves for other compression levels. Thus, JPEG compressing the input does not help the model train in fewer epochs.

IV. CONCLUSION

In this paper, we evaluated the potential of JPEG compression for pre-processing input data for classification. Through

TABLE II: Risk in JPEG-compressing input in comparison to increasing model size (number of parameters). The drop in performance with JPEG-compressing input images is significantly lower in comparison to the drop in performance observed with increasing model-size.

Sr. No	Data set	Small model with uncompressed input		Small model with compressed input		Large model with uncompressed input	
		Accuracy	Drop	Accuracy	Drop	Accuracy	Drop
1	DTD	0.6519	0	0.6545	0	0.5286	-0.1233
2	OxfordIITPet	0.9182	-0.0013	0.9169	-0.0013	0.8717	-0.0465
3	Caltech256	0.8551	-0.0087	0.8464	-0.0087	0.8379	-0.0172
4	CIFAR10	0.9617	-0.1095	0.8522	-0.1095	0.9617	-0.0182
5	SUN397	0.6815	-0.0039	0.6776	-0.0039	0.652	-0.0295

TABLE III: I/O benefits of JPEG compressing inputs for classification. We get anywhere between 65% to 86% reduction in storage space required for the input. In most data sets, this storage space reduction comes with negligible drop in performance.

Sr. No.	Data set	Storage size comparison			Accuracy comparison		
		Original Size	Compressed Size	Relative size of compressed to original	Original Accuracy	Compressed Accuracy	Relative performance of compressed to original
1	DTD	602.5 MB	133.8 MB	22.21%	0.6519	0.6545	100%
2	OxfordIITPet	759.6 MB	104.6 MB	13.77%	0.9182	0.9169	99.86%
3	Caltech256	1.1 GB	373.3 MB	33.14%	0.8551	0.8464	98.98%
4	CIFAR10	129.3 MB	44.3 MB	34.26%	0.9617	0.8522	88.61%
5	SUN397	37 GB	8.1 GB	21.89%	0.6815	0.6776	99.43%
6	CIFAR100	127.7 MB	44.3 MB	34.69%	0.83	0.64	77.11%

TABLE IV: No observable lift observed with using JPEG on noisy images for classification

Model name	Accuracy with Clean Input	Accuracy with Noisy Input	Accuracy with JPEG + Noisy Input
Autogluon	0.96	0.79	0.76

experiments on multiple data sets, we were able to validate that JPEG compression significantly reduces the storage requirement for input images - suggesting significant I/O advantages in production systems. Moreover, this comes at a negligible drop in classification performance. We also document that the relative risk to classification performance by JPEG-compressing the input is substantially less than training larger deep neural networks (keeping epoch-budget constant), thereby making JPEG preprocessing of input relatively low-risk. Finally, we also find that contrary to popular understanding, JPEG compressing images is not effective at dealing with noise for classification tasks neither is it helpful in training models faster.

REFERENCES

- [1] K. Takumi, K. Watanabe, Q. Ha, A. Tejero-De-Pablos, Y. Ushiku, and T. Harada, "Multispectral object detection for autonomous vehicles," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017, pp. 35–43.
- [2] S. A. Hicks, V. Thambawita, H. L. Hammer, T. B. Haugen, J. M. Andersen, O. Witczak, P. Halvorsen, and M. A. Riegler, "Acm multimedia biomedica 2020 grand challenge overview," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 4655–4658.
- [3] K. Liu and H. Ma, "Exploring background-bias for anomaly detection in surveillance videos," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1490–1499.
- [4] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [5] D. Le Gall, "Mpeg: A video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, no. 4, pp. 46–58, 1991.
- [6] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression," *arXiv preprint arXiv:1705.02900*, 2017.
- [7] G. Friedland, R. Jia, J. Wang, B. Li, and N. Mundhenk, "On the impact of perceptual compression on deep learning," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.
- [8] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *2016 eighth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2016, pp. 1–6.
- [9] —, "The impact of image quality on the performance of deep learning-based image recognition," in *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. IEEE, 2016, pp. 1–4.
- [10] Q. Zhao, S. Liu, and L. Karam, "The robustness of deep learning-based object detection algorithms to jpeg compression," *arXiv preprint arXiv:1708.01545*, 2017.
- [11] D. Wu, J. Xu, L. Li, S. Zhang, and W. Gao, "Compressed deep learning for image classification," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.
- [12] H. Talebi and P. Milanfar, "Nima: Neural image assessment," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42. IEEE, 2020, pp. 1202–1216.
- [13] J. Mueller, X. Shi, and A. Smola, "Faster, simpler, more accurate: Practical automated machine learning with tabular, text, and image data," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3509–3510.
- [14] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.