Adaptive Compression for Online Computer Vision:

An Edge Reinforcement Learning Approach

(Manuscript ID: TOMM-2020-0161.R2)

Response to the editor and reviewers

We would like to express our sincere gratitude to the editor and the reviewers for advising minor revision and providing valuable comments. Guided by the comments, we have tried our best to address the comments, and the amendments are highlighted in blue in the revised manuscript. In what follows, we include a detailed response to the reviewer and discuss how we have addressed the comments.

We hope that the revision has satisfactorily addressed all of the concerns raised by the review team. We look forward to hearing back from you soon.

With best wishes,

Sincerely,Zhaoliang He, Hongshan Li, Zhi Wang,

Shutao Xia, and Wenwu Zhu.

Response to Reviewer 1

**Comment 1** *How about the training process and performance of the designed RL algorithm? More details need to be further provided.*

**Response:** Thanks for your suggestion. Section 3.3 Reinforcement Learning-based Framework shows the main training process; Section 4.9.2 Training and Re-loading Latency shows the training latency and convergence condition; Section 4.4 Upload Image Size Overhead (Figure 8) presents the upload traffic load of the training and inference phase.

In the training phase, the upload traffic load is even higher than the conventional solution because AdaCompress has to upload the original image along with the compressed image to the cloud-end. Also, the accuracy is locked to 1 because AdaCompress uses the original image’s prediction label as the output result, and we add this description in Section 4.4 Upload Image Size Overhead.

**Comment 2** *There are still some typos and formatting issues such as the paragraph indentation problem.*

**Response:** Thanks for your careful check. We are sorry for our carelessness. We amend the paragraph indentation problem and carefully proofread the manuscript to clear out the writing problems.

Response to Reviewer 2

**Comment 1** *This paper presents an RL based approach to compress the images according to the required inference accuracy. The whole solution is reasonable. But I note that a similar problem has been studied in the literature. For example, "**Discernible Image Compression (MM 2020)" suggests conducting the compression by considering the downstream perception task as well. The authors took a different approach, i.e. RL based. But how to justify the necessity of the RL approach, instead of the end-to-end differentiable approach in the literature? Some discussion and experimental compression might be necessary here.*

**Response:** Thanks for your constructive comment. We agree with the missing discussion of our manuscript and have added a more detailed discussion in Section 2.2 Data Compression.

Firstly, our work is proposed earlier than theirs, and the preliminary version of this paper (AdaCompress: Adaptive Compression for Online Computer Vision Services) was published in ACM Multimedia 2019 (MM 2019).

Secondly, most studies improve the compression performance by optimizing the compression framework (e.g., the encoder/decoder, quantization function, entropy coding, and loss function), which would change the compression framework. For example, “Discernible Image Compression (MM 2020)” suggests using a powerful regularization to optimize the loss function by considering the downstream perception task，which changes the encoder-decoder network. In practical applications, for end-users, the compression framework always cannot be changed, and the cloud-based computer vision models are usually a black box. Without knowing details of the compression framework and cloud-end model structures, choosing the best compression level is a complicated decision-making problem because the optimal compression configuration is determined by confounding factors, including the characteristics of the input images and the model, the expected accuracy and image size, etc. To address this, we design a reinforcement learning agent that learns the optimal compression quality level to achieve an expected inference accuracy and upload image size, only from the online inference results.

Thirdly, compared with “Discernible Image Compression” is only applicative for end-to-end image compression methods, our framework is applicative for both well-known traditional compression methods (JPEG, WebP, H.265/HEVC, and H.266/VVC) and end-to-end (neural network-based learned) compression methods, which is detailed discussed in Comment 2 of Reviewer 2. Also, in the large-scale real-time online computer vision services, the traditional methods are more appropriate because their encoding-decoding speed is much faster than neural network-based learned compression methods.

**Comment 2** *In Line 40 on Page 9, the compression process seems to be taken as a black box. But it is unclear how that compression process takes the compression level as the input.*

*In addition, there are many image compression methods in the literature. Whether all these different techniques can flexibly take a compression level as the input? If not, the authors need to detail the concrete compression techniques used here. This point is closely related to the real-world applicability of the proposed framework.*

**Response:** Thanks for your suggestion. In our experiment, we choose the JPEG compression method as the compression framework. Each compression quality level represents a quantization table in the JPEG compression standard [1]. We use the PIL library (Python Imaging Library) [2] to complete the JPEG compression process, which can take the compression quality level as the input. We add this description in Section 4.1 Experiment Setup.

Our framework is applicative for different image compression techniques, both traditional compression methods and neural network-based learned compression methods. Existing traditional compression methods all have an optional compression level, e.g., JPEG and WebP have the compression quality level, H.265/HEVC and H.266/VVC have the quantization parameter (QP). Neural network-based learned image compression is a Lagrangian multiplier-based rate-distortion optimization, which commonly has a tradeoff parameter that controls the rate-distortion tradeoff [3-6]. Different values are corresponding to different bit rates. We can regard this tradeoff parameter as the compression level. These different compression techniques all can flexibly take a compression level as the input. We add this applicability discussion in Section 4.10 Applicability for Different Image Compression Methods.

**Comment 3** *In 3.3, the authors introduced the RL approach. However, the definition here is not rigorous enough. It is better to start with a clear MDP definition and introduce the concepts of state, action, reward within the problem context.*

**Response:** Thanks for your comment. We agree our definition is not rigorous enough and have added more detailed descriptions. Especially, we define a clear MDP formulation and introduce the concepts of state, action, and reward within the problem context. More details of the revision are provided in Section 3.3.

**Comment 4** *Images in the datasets are already compressed ones. Is that possible to evaluate the algorithm with the raw image without compression as the input?*

**Response:** Thanks for your comment. Most of the datasets used for computer vision-based services (e.g., recognition and detection tasks) are provided in the JPEG format. Also, most existing deep learning models only support input images with fixed sizes (e.g., 256 x 256) and need a large-scale validation dataset. However, the lossless image dataset (e.g., Kodak) is commonly a [PNG](http://www.libpng.org/pub/png/) format small dataset with high resolution.

The default compression quality level for JPEG is usually 75, and therefore we regard this as a reference value of the conventional benchmark (i.e., regard this reference image as the raw image).

**References:**

1. Wallace, Gregory K. "The JPEG still picture compression standard." IEEE transactions on consumer electronics 38.1 (1992): xviii-xxxiv.
2. Python Imaging Library. 2019. Image file formats. https://pillow.readthedocs.io/en/3.1.x/handbook/image-file-formats.html.
3. Ballé, Johannes, et al. "Variational image compression with a scale hyperprior." arXiv preprint arXiv:1802.01436 (2018).
4. Minnen, David, Johannes Ballé, and George D. Toderici. "Joint autoregressive and hierarchical priors for learned image compression." Advances in Neural Information Processing Systems. 2018.
5. Lee, Jooyoung, Seunghyun Cho, and Seung-Kwon Beack. "Context-adaptive entropy model for end-to-end optimized image compression." arXiv preprint arXiv:1809.10452 (2018).
6. Cheng, Zhengxue, et al. "Learned image compression with discretized gaussian mixture likelihoods and attention modules." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.