Image Coding for Machines with Objectness-based Feature Distillation

Kei Iino
Graduate School of FSE
Waseda University
Tokyo, Japan
iinokei@akane.waseda.jp

Shunsuke Akamatsu
School of FSE
Waseda University
Tokyo, Japan
s.akamatsu@akane.waseda.jp

Hiroshi Watanabe
Graduate School of FSE
Waseda University
Tokyo, Japan
hiroshi.watanabe@waseda.jp

Shohei Enomoto
NTT Software Innovation Center
Tokyo, Japan
shohei.enomoto@ntt.com

Akira Sakamoto
NTT Software Innovation Center
Tokyo, Japan
akira.sakamoto@ntt.com

Takeharu Eda
NTT Software Innovation Center
Tokyo, Japan
takeharu.eda@ntt.com

Abstract

Recently, the capability and popularity of automatic machine analysis of images and videos have grown rapidly. Consequently, the analysis of decoded images and videos by machines, rather than humans, is becoming more popular. This shift has led to a growing need for efficient compression methods that are optimized for machines instead of humans. In response to this demand, various methods of image coding for machines (ICM) have been developed. For training ICM models, distillation-based loss is often used. However, the valuable insights gained from the distillation methods in machine vision tasks have not been fully utilized yet. In this study, we propose an objectness-based feature distillation for ICM to improve rate-distortion (R-D) performance. We conducted experiments in object detection and instance segmentation tasks and confirmed that there was an improvement of up to approximately 1.5 points in mAP at the same rate.

Keywords: Image Coding, Coding for Machines, Learnable Compression, Knowledge Distillation

1. Introduction

These days, the performance of computer vision models has improved dramatically, leading to more instances where machines analyze decoded images and videos without human intervention. This trend has spurred active research about ICM [1,2], with standardization efforts such as JPEG-AI and Video Coding for Machines (VCM) underway. Modern learnable image codecs, often outperforming hand-crafted codecs in R-D performance, are increasingly used in the ICM domain [1,2]. When training ICM models, it's common to use distillation-based loss [1,2]. This makes sense in real-world deployments where access to labeled training data can be difficult or expensive. However, ICM distillation loss does not fully exploit the knowledge of distillation methods in machine vision tasks that have been studied for many years. In detection tasks, assigning higher weights to significant areas such as the foreground during the calculation of distillation loss is known to be an effective strategy. Therefore, in this research, we propose feature distillation, which is weighed by objectness, a component of the detector's output, aiming to enhance R-D performance in detection tasks.

2. Learnable Image Coding for Machines

Like typical learnable image coding for humans, an autoencoder-type CNN is used as a compression model. However, the decoded images are not meant for human viewing but are used as input for machine vision task models.

Generally, the parameters of the downstream task model are fixed, and only the compression model is trained. To train it, R-D loss is used:

$$L = R + \lambda D, \tag{1}$$

where R is an estimated rate of a bitstream, D is a distortion term to maintain task accuracy in the later stages, and λ is a Lagrange multiplier that controls the trade-off between rate and distortion. As mentioned before, considering real-world applications, it is common to use distillation-based loss as D without the use of labels. Specifically, a typical method involves using the mean squared error (MSE) between the features F obtained from original images X inputted into the task model and the features $\hat{F}$ obtained from compressed images $\hat{X}$ as the loss (Fig. 1) [1,2]:

$$D = \frac{1}{N}\sum_{i=1}^{N} MSE\left(F_i\,, \hat{F}_i\right), \tag{2}$$

where N is the number of the extracted features to calculate distillation loss.

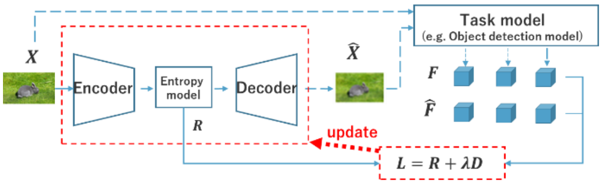

Figure 1: Feature distillation for training ICM model

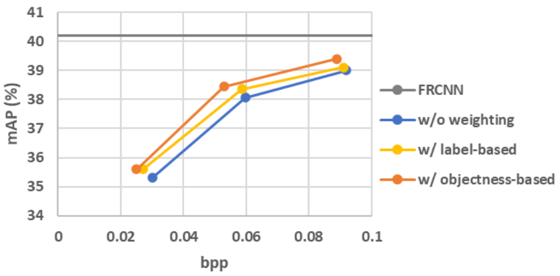

Figure 2: Object detection by FRCNN

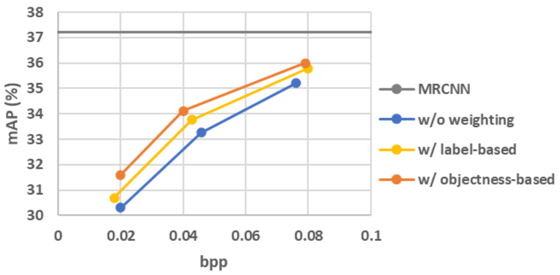

Figure 3: Instance segmentation by MRCNN

3. Proposed Method

In detection tasks, the method of feature distillation using weighted loss is known to be effective [3,4]:

$$D = \frac{1}{N}\sum_{i=1}^{N} W_i \cdot MSE\left(F_i\,, \hat{F}_i\right), \qquad (3)$$

where W is a weight map, usually created from ground truth (GT) labels. Some studies [3,4] have achieved improved performance by assigning greater weights to feature errors in the foreground compared to the background.

Inspired by this approach, we propose using objectness O as a weight map for feature distillation of ICM:

$$D = \frac{1}{N}\sum_{i=1}^{N} O_i \cdot MSE\left(F_i\,, \hat{F}_i\right). \qquad (4)$$

Objectness is the probability that an object exists in a proposed region of interest. Most object detectors, as well as segmentation models based on detectors, output this objectness or equivalent value [5,6,7]. When GT labels are used as weights, the compression model learns to preserve information about all objects, regardless of whether the downstream task model can detect them or not. On the other hand, by employing objectness instead of GT labels, the compression model is trained to disregard objects that are difficult to detect, and it leads to bit rate reduction. Furthermore, as acquiring labels for real-world data is often challenging, this approach is more practical and applicable in various settings.

4. Experiment

We tested the effectiveness of the proposed method in the two tasks: object detection and instance segmentation. In our experiments, we used a pre-trained compression model, cheng2020_attn [8] from comrpessAI [9] as a learnable ICM model. Specifically, we employed Faster R-CNN (FRCNN) [5] for object detection and Mask R-CNN (MRCNN) [6] for instance segmentation as downstream task models. During the experiment, feature maps {p2, p3, p4, p5} from the Feature Pyramid Network (FPN) [5,6] of the task model were used for feature distillation. In the proposed method, we used the objectness output of the Region Proposal Network (RPN) [5,6], for weighting loss. We used the COCO2017 dataset for our experiments, with a total of 50 epochs of training. At the 40th epoch, the learning rate was reduced by a factor of 0.1. The initial learning rate was 1e-4 and the Adam was used as an optimizer.

The results of the R-D curves of each task are shown in Fig.2 and 3. From these figures, it is clear that the R-D performance is improved by using the proposed method. Also, the yellow line in the figure represents the result of weighting using GT label. This confirms that weighting using objectness is superior to using GT labels.

5. Conclusion

Our research focused on feature distillation used in ICM model training. In the field of computer vision, there are a variety of techniques for feature distillation. In detection tasks, feature distillation by weighting with GT labels is often used. In this study, we proposed a feature distillation technique that uses weighting with objectness instead of GT labels. The proposed method not only outperforms traditional distillation but also outperforms GT label weighting approach.

References

[1] A. Harell et al., “Rate-distortion theory in coding for machines and its application,” arXiv preprint:2305.17295, (2023).

[2] H. Choi et al., “Scalable image coding for humans and ′ machines,” IEEE Transactions on Image Processing, vol. 31, pp. 2739–2754 (2022).

[3] Borui Zhao et al., “Decoupled knowledge distillation,” Proc. CVPR. IEEE/CVF, pp. 11953–11962, (2021).

[4] Y. Zhendong et al., “Focal and Global Knowledge Distillation for Detectors,” Proc. CVPR. IEEE/CVF, pp. 4643-4652, (2022).

[5] S. Ren et al., “Faster r-cnn: Towards realtime object detection with region proposal networks,” Proc. NeurIPS, vol. 28. (2015).

[6] K. He et al., “Mask R-CNN,” Proc. ICCV, pp. 2961–2969 (2017).

[7] J. Redmon et al., “YOLOv3: An incremental improvement,” arXiv preprint:1804.02767, (2018).

[8] Z. Cheng, et al, “Learned image compression with discretized gaussian mixture likelihoods and attention modules,” Proc. CVPR. IEEE/CVF, (2020).

[9] J. Begaint et al., “CompressAI: a PyTorch library and evaluation platform for end-to-end compression research,” arXiv preprint:2011.03029, (2020).