Image Coding for Machines with Omnipotent Feature Learning

Ruoyu Feng[1*] Xin Jin[2*] Zongyu Guo[1] Runsen Feng[1] Yixin Gao[1]
Tianyu He[3] Zhizheng Zhang[3] Simeng Sun[1] Zhibo Chen[1,†]

[1] University of Science and Technology of China
[2] Eastern Institute of Advanced Study
[3] Microsoft Research Asia, Beijing, China
ustcfry@mail.ustc.edu.cn jinxin@eias.ac.cn
chenzhibo@ustc.edu.cn

Abstract. Image Coding for Machines (ICM) aims to compress images for AI tasks analysis rather than meeting human perception. Learning a kind of feature that is both general (for AI tasks) and compact (for compression) is pivotal for its success. In this paper, we attempt to develop an ICM framework by learning universal features while also considering compression. We name such features as omnipotent features and the corresponding framework as Omni-ICM. Considering self-supervised learning (SSL) improves feature generalization, we integrate it with the compression task into the Omni-ICM framework to learn omnipotent features. However, it is non-trivial to coordinate semantics modeling in SSL and redundancy removing in compression, so we design a novel information filtering (IF) module between them by co-optimization of instance distinguishment and entropy minimization to adaptively drop information that is weakly related to AI tasks (*e.g.*, some texture redundancy). Different from previous task-specific solutions, Omni-ICM could directly support AI tasks analysis based on the learned omnipotent features without joint training or extra transformation. Albeit simple and intuitive, Omni-ICM significantly outperforms existing traditional and learned-based codecs on multiple fundamental vision tasks.

Keywords: Image coding for machines, Self-supervised learning, Information filtering.

1 Introduction

In the big data era, massive images and videos have become an indispensable part of people's production and life. As an important industrial technology, lossy image compression aims to save storage resources and transmission bandwidth by preserving the most critical information. In the past decades, the traditional image and video coding standards such as JPEG [67], JPEG2000 [58], AVC/H.264 [68], HEVC/H.265 [64], VVC/H.266 [6] have significantly improved

* First two authors contributed equally.
† Corresponding author.

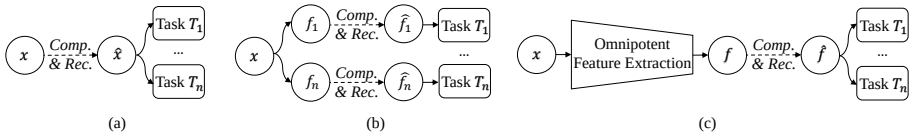

Fig. 1: Comparison of three branches for image coding for machines (ICM). They are different from each other w.r.t the object to be compressed and the characteristics of task-specific or not. (a): Codecs in this branch support downstream tasks by inputting the decompressed images. (b): One-to-one features-based ICM solution, the decompressed features of corresponding tasks are input to the task models. (c): With the proposed **omnipotent feature** f extracted and compressed first, all the downstream tasks could complete the inference based on the decompressed feature $\hat{f}$.

the coding efficiency. Recently, with the fast development of deep neural networks, learned-based image compression codecs [3,4,52,20,38,43,42,50,51,14,69] have achieved a great success. They have potentials to become the next-generation image compression standards due to the high performance and applicability compared to traditional hand-craft codecs. Meanwhile, deep neural networks has demonstrated their potential in various computer vision tasks, *e.g.*, object detection [61,59,60,45], instance segmentation [32,47,5], semantic segmentation [48,1,9,10], pose estimation [32,54]. We can anticipate that more and more data transmitting on the Internet would be consumed by machines for intelligent analysis tasks.

However, all the image compression methods mentioned above aim at saving transmitting costs while improving the reconstruction quality for human perception. When facing AI tasks analysis, existing image coding methods (even for the deep learned-based) are still questionable, regarding whether it can encode images efficiently, especially in application scenarios for big data. To facilitate the performance and efficiency in terms of high-level machine vision tasks that act on lossy compressed images, lots of research efforts have been dedicated to a new problem of image coding for machines (ICM) [26,41], which aims to compress the source image for supporting the intelligent analysis tasks. The discrepancy between human-perception oriented metric (*e.g.*, mean square error (MSE), multi-scale structured similarity (MS-SSIM)) and AI task metric (*e.g.*, classification accuracy) makes ICM particularly different from the existing compression schemes.

For ICM, there mainly exist solutions of two branches. Fig. 1(a) shows the first branch that the compressed image is sent into the downstream task model for intelligent analytics. Codecs in this branch are typically designed based on a heuristic RoI (Region of Interest) bit allocation strategy [63,21,7,36] or joint optimization for image reconstruction with a task-specific constraint in an end-to-end manner [41]. This branch has two weaknesses that the image reconstruction brings more computational burden because images have to be reconstructed for

subsequent intelligent analysis and there exists a new trade-off between texture fidelity and semantics integrity. The second branch is a one-to-one feature-based ICM framework [16,17,62,2]. As shown in Fig. 1(b), works of this branch tend to compress the features extracted from images for transmission efficiency. Depending on the reconstructed features, the downstream tasks could directly complete the corresponding intelligent analysis. But, such a scheme that one compressed feature can only be used to support one specific AI task lacks generalization and flexibility, thus is difficult to be applied to practical applications.

To solve the problems mentioned above, and motivated by the urgent requirements for a generalized ICM solution, in this paper, we go beyond previous pipelines and introduce a unified framework for ICM by exploring the "common knowledge" of different AI tasks. More precisely, a novel ICM framework, termed Omni-ICM, is designed based on learning omnipotent features for machines, as shown in Fig. 1(c). The omnipotent features are expected to be general for different intelligent tasks and compact enough that only contain the semantics relevant information. They can be regarded as new representations "seen" by machines. To achieve the omnipotent feature learning, we borrow ideas from the popular contrastive learning that has been proved could learn general and transferable visual representations [31,11,8,28,13], and integrate it into the image coding pipeline. However, directly compressing the features learned by contrastive objective has no obvious advantages than compressing the original images directly [15,16,18], that's because these features typically keep lots of irrelevant redundant information with no explicit constraint on information entropy.

To tackle this issue, we further design an Information Filtering (IF) module to smartly discard the redundant information for analytics before compression, so as to encourage learned representations to be sparse and compact. Basically, the IF module comprises an encoder, a decoder, and an entropy estimation model, and is optimized with contrastive loss and entropy minimization constraint. In this way, IF module learns to preserve semantic-wise information and filter out redundant ones, acting as a bridge to connect contrastive training and compression. After that, with a learned-based feature compressor, the learned omnipotent features are compressed and reconstructed in the feature latent space, enabling it to be directly input to downstream task models without pixel-level reconstruction. Moreover, compressing such omnipotent features makes it more applicable to the codec standardization, which could support for a wide range of downstream AI tasks, even for the unknown ones. Such generalization ability and flexibility are the key points of our Omni-ICM framework, which are often neglected by the existing ICM solutions.

Extensive experiments show that Omni-ICM outperforms the state-of-the-art image compression methods by significant margins w.r.t the bitstream saving and task performance, on multiple intelligent tasks, including object detection, instance/semantic/panoptic segmentation, and pose estimation.

2 Related Work

2.1 Image Compression

Traditional Codec. Traditional hand-craft image codecs typically consist of intra prediction, transformation, quantization, and entropy coder. The popular image coding standards have kept evolving, *e.g.*, JPEG [67], JPEG2000 [58], AVC [68], HEVC/H.265 [64], VVC/H.266 [6]. However, these codecs cannot be optimized in an end-to-end manner, thus lack of flexibility and scalability to support different objectives, such as MS-SSIM and classification accuracy.

learned-based Codec. The success of deep learning techniques significantly promotes the development of learned-based codecs. Toderici *et al.* [66] apply a recurrent neural network (RNN) to end-to-end image compression, achieving a comparable performance with JPEG. Ballé *et al.* [3] further propose an end-to-end framework based on nonlinear transformation, generalized divisive normalization (GDN), noise-relaxed quantization, and their method outperforms JPEG 2000. Then a variational model with hyperprior is introduced to parameterize latent distribution with a Gaussian distribution in [4]. Some recent works have improved image compression from the aspects of entropy coding [52,53,20,29,39] and quantization [30,73]. However, the optimization objectives of these methods are pixel-level metrics that designed for visual fidelity, *e.g.*, MSE, MS-SSIM. The discrepancy between pixel-level distortion and semantic-level distortion leads to the failure of above methods when tackling ICM tasks. But, they provide basic techniques to develop effective ICM solutions to handle this new problem.

Image Coding for Machine. ICM [26,34] aims to compress and transmit the source image for machines to support intelligent tasks. Based on the heuristic prior knowledge of foreground matters more for intelligent analysis, [7,36,44] merge the ROI (Region of Interest) based bit allocation strategy into the traditional codec for intelligent analytics. For learned-based codecs, Le *et al.* [41] propose an image compression system that jointly optimizes models for object detection and reconstruction. Codevilla *et al.* [22] also optimize both the intelligent task and the reconstruction task, and the difference is that the optimization of the intelligent task directly takes the latent variable features as input. However, the trade-off between semantic fidelity and pixel fidelity limits their respective performance. Thus, [35,72] introduce scalable coding to coordinate the compression for high-level information and pixel-wise texture. Singh *et al.* [62] explore to compress features instead of images for intelligent tasks by optimizing the task objective along with rate loss. Nevertheless, such schemes can only support a few tasks and are not general enough. The recent work of SSIC [65] structures the bitstream according to the object category and thus achieves a task-aware compression for downstream analytics. Differently, in this paper, we aim to design a unified framework for ICM by learning a kind of general and compact features and directly support a wide range of intelligent tasks.

2.2 Self-supervised Representation Learning (SSL)

Self-supervised learning [37,74,57,55] is proposed to learn general representations for downstream tasks by solving various pretext tasks on large-scale unlabeled

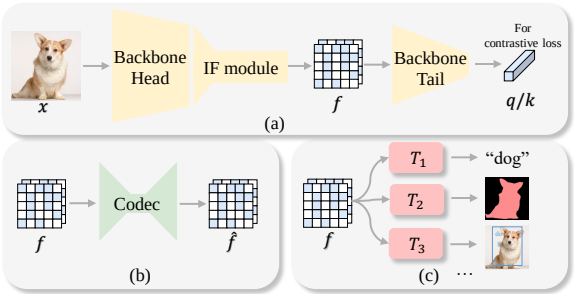

Fig. 2: Three stages in our Omni-ICM framework. (a) Omnipotent feature learning. We optimize the whole network with the contrastive loss and entropy constraint by the IF module. (b) Omnipotent feature compression. A feature compressor is trained for omnipotent feature compression, with all parameters fixed except the codec. (c) Omnipotent feature deployment. Our Omni-ICM can easily support different downstream tasks by fine-tuning the backbone tail with omnipotent features as input.

datasets. Contrastive learning is one of them and its pretext task is minimizing feature distances from the same group and maximizing feature distances from different groups with contrastive loss. Recently, the siamese network based contrastive learning methods [31,11,8,28,13] have drawn lots of attention. Among them, MOCO [31] is the first work that outperforms the supervised ImageNet pre-training on several downstream tasks, which shows its strong ability for general representations learning. More specifically, MOCO designs a dynamic queue to store negative samples features and uses a momentum update mechanism to optimize the model progressively. Inspired by that, we propose to employ SSL to learn omnipotent features for compression, so that further support heterogeneous intelligent tasks for ICM.

3 ICM with Omnipotent Feature Learning

3.1 Overview of Omni-ICM Pipeline

We propose a new concept of omnipotent feature learning for image coding for machines, and correspondingly design a unified framework (Omni-ICM) based on it. As shown in Fig. 2, the whole framework of Omni-ICM consists of three stages: (a) omnipotent feature learning, (b) omnipotent feature compression, and (c) omnipotent feature deployment.

For the first stage, we employ a contrastive learning pipeline while also giving consideration to compression efficiency, enabling the learned features to be both semantically preserved and compact. More specifically, to coordinate the preserving of the semantics and the discarding of the semantic-irrelevant redundancy, we design an additional Information Filtering (IF) module and optimize the whole network with an instance-contrastive loss under entropy constraint.

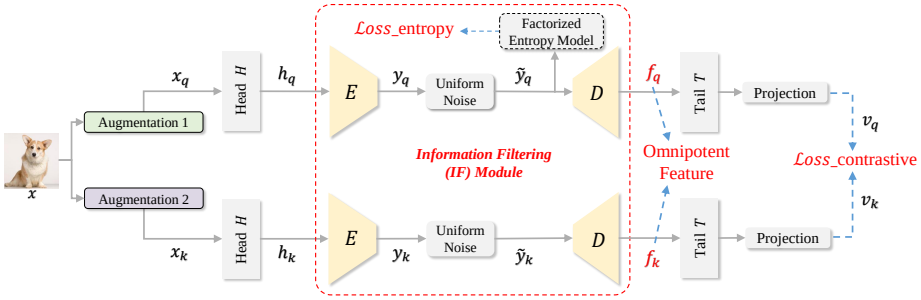

Fig. 3: Architecture of omnipotent feature learning. We use a pair of query and key for simpler illustration. By maximizing the similarity of different views of an image under entropy constraint, the network learns to discard semantic-redundant information and keep critical ones. After training, f is the omnipotent feature we need.

After that, the obtained omnipotent features, which are compact and general, are "seen" by machines as an alternative for original images. To compress and transmit the omnipotent features, we additionally train a feature codec. Finally, the downstream tasks supporting are achieved by fine-tuning the backbone tail. Note that, the backbone head and the proposed IF module are fixed in this stage. We describe each stage in detail in the following subsections.

3.2 Stage 1: Omnipotent Feature Learning

Basic Network Architecture. Considering that the learned omnipotent features will be taken for a wide range of AI tasks analytics, *e.g.*, object detection [45], semantic segmentation [75], we extract the omnipotent feature f with a $4\times$ down-sampling factor to promise the integrity of content structure and object spatial layout. Specifically, as shown in Fig. 3, a commonly used backbone (such as ResNet-50) is split into two parts, namely backbone head and backbone tail, dotted as H and T. In a ResNet-50, the backbone head comprises the stem layer and layer1, and the backbone tail comprises layer2∼layer4.

Data Augmentation and Feature Extraction in Backbone Head. As illustrated in Fig. 3, at the omnipotent feature learning stage, two views of an image x_q and x_k are first generated by different augmentations. For clarity, we describe the query generation process for x_q at first. x_q is fed into the backbone head H, obtaining an $4\times$ down-sampling feature with a size of $\frac{H_q}{4} \times \frac{W_q}{4} \times C$, where H_q, W_q are the height and width of x_q, C means the channel numbers:

$$h_q = H(x_q). \tag{1}$$

Information Filtering (IF) Module. Importantly, the representation directly generated by the backbone head is not suitable for ICM, because it still contains lots of semantic-irrelevant information (see the third column of Fig. 9). Thus, we design an additional information filtering (IF) module between the backbone

head and tail, to simultaneously achieve the preservation of semantic information and the dropout of irrelevant information. The IF module consists of an encoder, a factorized entropy model, and a decoder denoted as E, F, D. To drive the IF module to learn to filter out the redundant information, an entropy constraint is enforced on it.

Formally, h_q is first fed into the encoder E of IF module with 8× downsampling, obtaining a latent variable y_q with the size of $\frac{H_q}{32} \times \frac{W_q}{32} \times C_y$, C_y represents the channel numbers of y_q:

$$y_q = E(h_q). \tag{2}$$

Then, a factorized entropy model F estimates the entropy of y_q through adding an additive uniform noise[3] on it to get the derivative $\tilde{y}_q$, formulated as:

$$p_{\tilde{y}_q|\phi_o}(\tilde{y}_q|\phi_o) = \prod_i (p_{y_q|\phi_o}(\phi_o) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\tilde{y}_q), \tag{3}$$

where $\boldsymbol{\phi_o}$ represents the parameters in H and E. And, the entropy loss is:

$$\mathcal{L}_e = \mathbb{E}[-\log_2(p_{\tilde{y}_q|\phi_o}(\tilde{y}_q|\phi_o))]. \tag{4}$$

Finally, $\tilde{y}_q$ is fed into the decoder D of IF module, obtaining the feature f_q with the same size as the input of IF module, *i.e.* $\frac{H_q}{4} \times \frac{W_q}{4} \times C$.

Backbone Tail and Projection Layer. With the feature f_q generated by the IF module, the backbone tail and a projection layer are employed to map the feature to the space where contrastive loss is applied. Specifically, the projection layer is an MLP with one hidden layer. This procedure can be formulated as:

$$q = W^{(2)}\sigma(W^{(1)}(T(D(\tilde{y}_q)))), \tag{5}$$

where σ is a ReLU non-linearity transformation, $W^{(1)}$ and $W^{(2)}$ are fully connected layers, $q \in \mathbb{R}^d$.

Generation of Keys. x_k is obtained by the other augmentation from the same image. The key x_k and the query x_q together construct a positive pair. For simplicity, we use the same notation in Section 3.2 here. This procedure can be formulated as:

$$y_k = E(H(x_k)), \tag{6}$$

$$k_+ = W^{(2)}\sigma(W^{(1)}(T(D(\tilde{y}_k)))), \tag{7}$$

where $\tilde{y}_k$ comes from y_k by adding the additive uniform noise, and $k_+ \in \mathbb{R}^d$, denotes the positive sample. The negative samples come from different images, denoted as$\{k_-\}$, are provided by the queue coming from the previous iterations [31]. Following the setting in MOCO [31], the branch of keys is the momentum-updated one of the branch of queries.

Total Optimization Objectives. For the contrastive loss, InfoNCE [56] is employed to pull q close to k_+ while pushing it away from other negative keys:

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+/\tau)}{\exp(q \cdot k_+/\tau) + \sum_{k_-} \exp(q \cdot k_-/\tau)}, \tag{8}$$

where τ denotes a temperature hyper-parameter as in [71]. The overall optimization function is written as:

$$\mathcal{L} = \mathcal{L}_q + \alpha \mathcal{L}_e, \tag{9}$$

where a Lagrange multiplier α is a fixed value that determines the trade-off between entropy and semantic integrity. Note that, the added additive noise is only a transitional component for entropy estimation in the omnipotent feature learning stage, and is discarded in the next two steps, *i.e.* omnipotent feature compression and deployment.

3.3 Stage 2: Learned-based Feature Compression

Similar to lossy image compression, the goal of lossy feature compression is simultaneously minimizing the size of bitstream and the distortion between f and $\hat{f}$. Such objectives can be formulated as minimizing $R + \lambda D_C$ (here we use D_C to distinguish the D in IF module), where the Lagrange multiplier λ controls the trade-off between the rate R and the distortion D_C in feature level. R denotes the rate of compressed feature and D_C represents the distortion between f and $\hat{f}$. Since quantization is non-differentiable, the additive uniform noise [3] is added to the latent variables during training for approximately rate estimation, which alters quantization to be differentiable. And, after quantization, the entropy coding is performed on latent variables y to encode it into bitstream losslessly. Entropy coding here can be Huffman coding or arithmetic coding. Finally, for the omnipotent feature reconstruction, the decoder tend to reconstruct omnipotent features from $\hat{y}$. The R-D (rate-distortion) loss function can be written as:

$$\mathcal{L}_{rd} = \mathbb{E}[-\log_2(p_{\hat{\boldsymbol{y}}|\boldsymbol{\psi}}(\hat{\boldsymbol{y}}|\boldsymbol{\psi}))] + \lambda \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} (f_{x,y} - \hat{f}_{x,y})^2, \tag{10}$$

where W and H denotes the width and height of features.

Moreover, since the features are compressed to handle downstream tasks better, we further protect its semantic fidelity in a deeper feature level. Particularly, the omnipotent feature f and its reconstructed one $\hat{f}$ are passed through the backbone tail in the omnipotent feature learning stage, *i.e.* layer2∼layer4 in a normal ResNet. And then, the Euclidean distance is calculated between those two deeper feature representations of f and $\hat{f}$ to construct this loss:

$$\mathcal{L}_f = \sum_{i=2}^{4} \lambda_i \frac{1}{W_i H_i} \sum_{x=1}^{W_i} \sum_{y=1}^{H_i} (\phi_i f_{x,y} - \phi_i \hat{f}_{x,y})^2, \tag{11}$$

where W_i and H_i are widths and heights of feature maps, ϕ_i means a differentiable function, hyperparameter λ_i controls the importances of distortions in different depths. The overall loss function of feature compression is given by:

$$\mathcal{L}_{com} = \mathcal{L}_{rd} + \mathcal{L}_f. \tag{12}$$

Practically, we design the neural network for omnipotent feature compression, which is derived from the Mean & Scale (M&S) Hyperprior model [52], and discretized Gaussian Mixture Likelihoods (GMM) entropy model [20].

Last but not least, there are two autoencoders in our pipeline, however, with different architectures, implementations, and functions. The first autoencoder in IF module is optimized with both contrastive loss and entropy constraint, without hard quantization operation in practice, acting as an information filter. The other autoencoder is used for feature compression, with hard quantization in practice. Detailed architectures are reported in **Supplementary**.

3.4 Stage 3: Feature Deployment and Task Supporting

After the omnipotent feature learning, the source data for machines has changed from images to omnipotent features. Therefore, the task models are trained with the learned omnipotent features f and are evaluated with the reconstructed omnipotent features $\hat{f}$, to finally support the AI tasks. Formally, only the backbone tail is fine-tuned for downstream tasks supporting, and the weights obtained in the omnipotent feature learning stage are used for a better initialization.

4 Experiments

4.1 Datasets

The training for both omnipotent feature learning and feature compression is conducted on the training set of the ImageNet [25] dataset, which contains ~ 1.28 million images of 1000 classes. After the training of feature extraction and compression, we evaluate the transferability of the learned omnipotent features to downstream tasks on PASCAL VOC [27], MS COCO [46] and Cityscapes [24]. PASCAL VOC and MS COCO are the widely-used datasets for dense prediction tasks, *e.g.*, object detection, instance segmentation. Compared with PASCAL VOC, MS COCO is larger and more challenging (more complicated scenes, more objects per image, and more categories to be predicted). Cityscapes is a fundamental and challenging dataset for semantic segmentation, which contains 5000 high-quality images with the pixel-level annotations (2975, 500, and 1525 for the training, validation, and test sets respectively).

4.2 Implementation Details

Omnipotent feature learning. With ResNet-50 [33] as the basic architecture, the IF module takes the output of backbone head as input to obtain the omnipotent feature. In the omnipotent feature learning stage, the momentum update from one encoder to another is set to 0.999 and the dictionary size is set to 65536. Temperature in Eq. (8) is set to 0.2. The data augmentation operations and the use of MLP projection head are same as the previous contrastive learning related works [12,28,11,13,31]. Besides, we load the weights that pre-training 800 epochs with MOCO-v2 [31] to initialize the backbone head and backbone tail, and then keep all parameters fixed except the IF module for a stable training at the first 10 epochs. After that, all the parameters are optimized together for another 200 epochs. We adopt SGD as the optimizer with weight decay and momentum set as 10^{-4} and 0.9. The batch size is 256 and the learning rate is 10^{-3}. α in Eq. (9) is experimentally set to 0.1.

Omnipotent Feature Compression. We train the omnipotent feature compressor model for $400,000$ iterations with batch size of 32. We employ the

Adam [49] optimizer, where the learning rate is set to be 5×10^{-5}. Data augmentation is 256×256 random cropping. λ in Eq. (10) is set to 2048, and $\lambda_2, \lambda_3, \lambda_4$ in Eq. (11) are set to 512, 256, 128 respectively. Feature codecs with different rates are obtained by multiplying λ, λ_2, λ_3, and λ_4 by a same coefficient.

4.3 Effectiveness and Superiority of Omni-ICM

Evaluation Protocol. We evaluate the generalization of omnipotent features on different fundamental intelligent tasks by fine-tuning the backbone tail. Challenging and popular datasets are adapted for different tasks, *i.e.* VOC object detection, COCO object detection, COCO instance segmentation, COCO pose estimation, Cityscapes semantic segmentation, and Cityscapes panoptic segmentation. Experiments for Cityscapes semantic segmentation are implemented in [23] and others are implemented in [70]. To evaluate the rate-distortion performance, the rate is measured by the bits per pixel (bpp), which is calculated by dividing the size of the feature bitstream by the number of pixels in the original image, and the distortion here represents metrics of different AI tasks.

Comparison Approaches. We mainly compare our Omni-ICM with the most advanced traditional codecs (HEVC [64], VVC [6]) and a learned-based compression method[20]. To ensure the fairness of comparison, we use the pre-trained model that has trained for 800 epochs on ImageNet [12] as the initial weights and fine-tunes it on each task to get the well-trained networks for comparison, which is consistent with the operations taken by the current SOTA representation learning method, MOCO [31]. Then during evaluation of compared approaches, reconstructed images are input into these networks to obtain the final results. Our method and the compared methods follow the same training schedule for fine-tuning downstream tasks. Besides, in order to better understand the results, we provide results with uncompressed images or features performing intelligent tasks, which can be seen as baselines. We also report down-stream task performances with supervised pre-training in **Supplementary**.

Object Detection on PASCAL VOC. When evaluating on VOC object detection, we follow the common protocol that fine-tuning a Faster R-CNN detector (C4-backbone) on the VOC `trainval07+12` set and testing on the VOC `test2007` set. The image scale is in $[640, 800]$ pixels during training and is 800 at inference as default. Note that the image resolution has changed before inputting into the task model. For the fairness of comparison, we don't perform any resizing operations on the features, and we regard the original image as the source data to be compressed so that we calculate the rate by dividing the size of the bitstream file of feature by the number of pixels of the original image. Other tasks that need resizing during preprocessing all obey this setting, *i.e.* instance segmentation, pose estimation. Fig. 4 (left) shows the results of detection. Our method achieves the best performance (lower rate, higher precision).

Semantic Segmentation on Cityscapes. For semantic segmentation, an FCN-based structure is used. We train task networks on the `train_fine` set which consist of 2975 images for 80k iterations, and evaluate on the `val` set. Results are shown in Fig. 4 (right). Similarly, our method is also the best scheme.

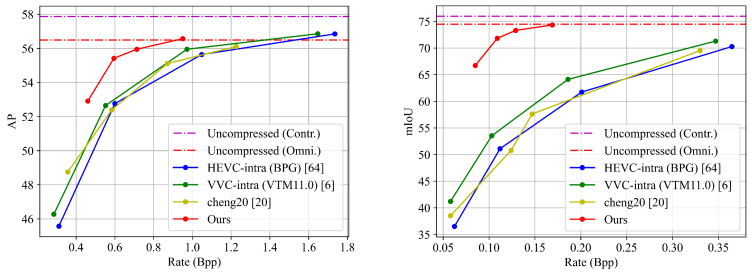

Fig. 4: **Object detection mAP on PASCAL VOC (left) and semantic segmentation mIoU on Cityscapes (right) under different bitrates.** We compare our method with two traditional codecs HEVC-intra [64], VVC-intra [6], and one learned-based codec [20].

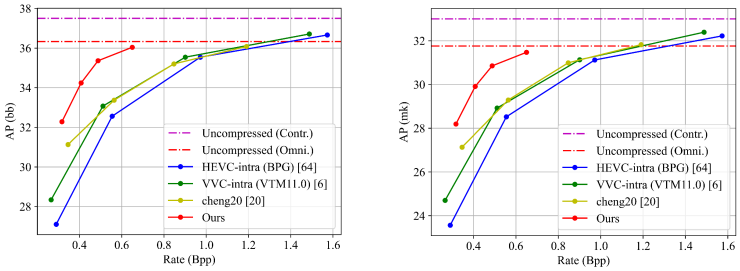

Fig. 5: **Object detecion and instance segmentation on MS COCO.** The metrics here include mean bounding box AP (AP^{bb}) and mask AP (AP^{mk}).

Object Detection and Instance Segmentation on MS COCO. Following the setting in [31], we evaluate object detection and instance segmentation by fine-tuning a Mask R-CNN detector (C4-backbone) on COCO `train2017` split with the standard 1× schedule and evaluating on COCO `val2017` split, with BN tuned and synchronized across GPUs. The image scale is in [640, 800] pixels during training and is 800 at inference as default, same as that for PASCAL VOC. The comparison is shown in Fig. 5. Our method also achieves the best performance, and significantly outperforms the other codecs.

More Downstream Tasks. Fig. 6, 7 show results on more downstream tasks:

COCO pose estimation: Mask R-CNN (with R50-FPN) is fine-tuned on COCO `train2017` and evaluated on `val2017`. The schedule is 1×. Results are illustrated in Fig. 6. Although Omni-ICM is better than other methods, however, there exists an obvious gap (more than 2 points in both person detection and keypoint detection) between the best performance at high bitrate. This also indicates the superiority of our method at lower bitrates.

Cityscapes panoptic segmentation [40,19]: Panoptic-deeplab [19] is used for this task. We train task networks on the `train_fine` set for 90k iterations, and evaluate on the `val` set. Results of PQ, mIoU, and AP are reported for panoptic

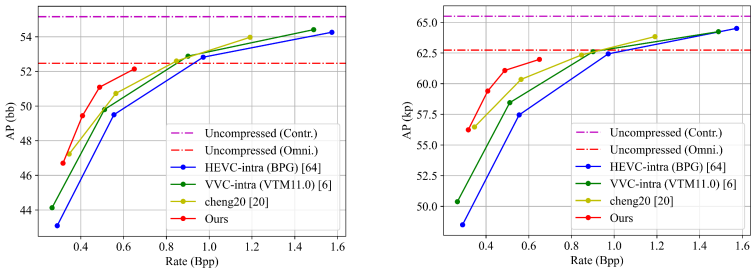

Fig. 6: **Pose estimation on MS COCO.** Results of person detection (AP^{bb}) and keypoint detection (AP^{kp}) are illustrated.

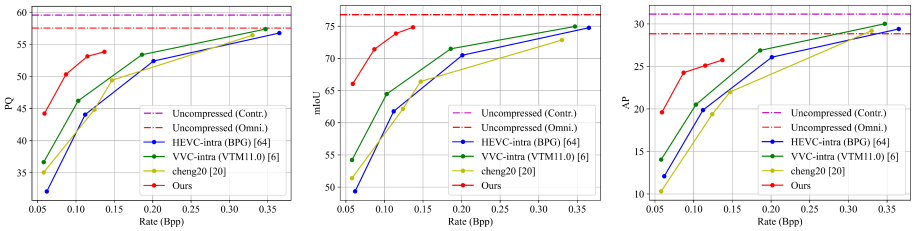

Fig. 7: **Panoptic segmentation on Cityscapes.** PQ, mIoU, and AP are reported. PQ is the metric of panoptic segmentation which measures the performance for both stuff and things in a uniform manner, mIoU is the metric of semantic segmentation, and AP is the metric of instance segmentation.

segmentation in Fig. 7. The performance of mIoU is similar to Fig. 4 (right). We can observe that our method achieves the better R-D performance, which means it can use less bits to achieve higher task performance.

Discussion. For the case of image coding for machines (ICM), Omni-ICM outperforms the most advanced hand-craft traditional codecs and a learned-based codec by remarkable margins on 6 fundamental intelligent tasks. Besides, we also observe some hidden limitations. Results in Fig. 6 and Fig. 7 show the potential performance gaps at the highest bitrate. We speculate that this is caused by two reasons. The first one is the discrepancy between datasets, the ImageNet is mainly composed of images with a single conspicuous target in natural scenes, while the number of targets in MS COCO and Cityscapes is diversified, and the scales of targets are also various. The second reason is that training by instance discrimination [31,11,56] forces the model to focus more on the conspicuous part of the image, which is not conducive to the preservation of local semantic information that occurs frequently in the above two datasets. In addition, we also compare our method with SOTA ICM-related methods and report the results in **supplementary**.

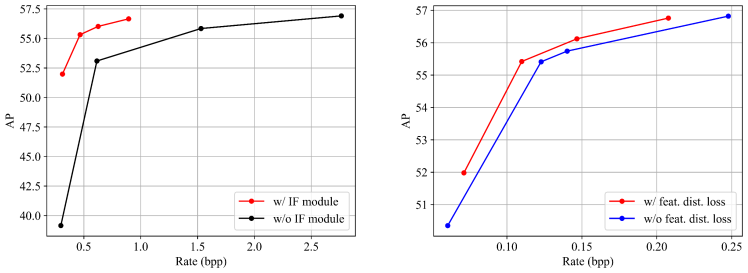

Fig. 8: Ablation studies on **IF module** (left) and **feature level distortion loss** (right), respectively.

4.4 Ablation Study

We implement ablation studies by pre-training on ImageNet and fine-tuning on VOC0712 object detection, as introduced in 4.3.

Study on IF module. The first graph in Fig. 8 illustrates the result that validate the contribution of IF module. For the case without IF module, the features output by layer1 of the ResNet-50 network pre-trained by contrastive learning are employed for task supporting and compression. Thus, we fix parameters in stem layer and layer1, and then fine-tunes the task model on PASCAL VOC detection. A feature codec with the same architecture and training schedule as that in Section 3.3 is trained for feature compression. As we can see, in the absence of IF module, compressing features directly can achieve satisfying performance with low coding efficiency. However, our Omni-ICM can achieve comparable performance with much lower bitrate.

Feature level distortion loss. The second graph in Fig. 8 presents the ablation study about feature-level distortion in Eq. (11). It indicates that the loss of feature level distortion helps protect semantic information.

4.5 Vision Analysis and Insights

Reconstruction Results. To better understand the functionality of the IF module, we additionally train two decoders with MSE loss to visualize the reconstruction results of features before and after IF module, *i.e.* h and f. As shown in Fig. 9, images reconstructed from h contain slight color difference, and textures are relatively complete. But images reconstructed from f suffer obvious color difference and texture loss. It indicates that IF module drops out some color information and texture information that has a slight influence on intelligent analytics. Details of reconstruction decoders are reported in **Supplementary**.

Bit allocation Map. As is illustrated in Fig. 10, we also visualize the bit allocation maps in IF module and that in the learned-based codec [20] optimized with MSE loss. Learned-based codec tends to focus on areas with large, irregular, and complex textures, *e.g.*, walls, water surfaces, rocks, and eaves. But our IF module pays less attention to the texture details in the image and more attention to the objects, which is crucial for the understanding of images.

Fig. 9: **Reconstruction of features before and after IF module.** Numbers on the top of the crop images indicate PSNR (dB) / MS-SSIM of an entire image.

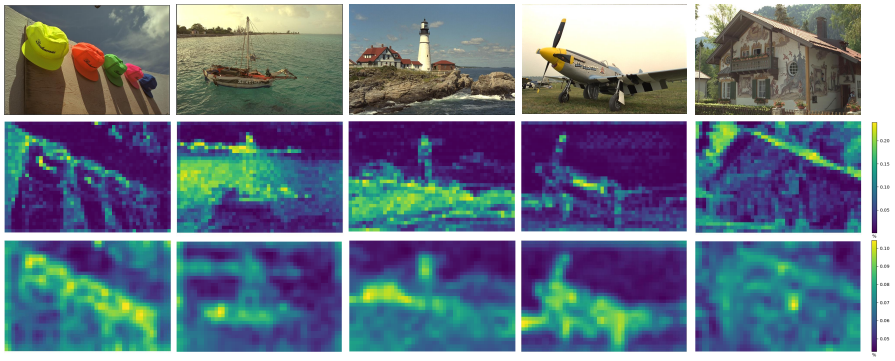

Fig. 10: **Bit allocation maps in learned-based codec [20] (second line) and our IF module (third line), respectively.** The first line is ground truth.

5 Conclusion

We presented a novel framework for image coding for machines (Omni-ICM) based on extracting and compressing a general and compact feature, dubbed omnipotent feature. The omnipotent feature is learned by elegantly combining the contrastive learning and entropy constraint through a new IF module, which coordinates semantics modeling and redundancy removing in our framework by adaptively filtering information that weakly related to AI tasks. Extensive experiments show an outstanding performance of our proposed Omni-ICM framework compared to the SOTA traditional and learned-based approaches.

Acknowledgement

This work was supported in part by NSFC under Grant U1908209, 62021001 and the National Key Research and Development Program of China 2018AAA0101400.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. TPAMI **39**(12), 2481–2495 (2017)
2. Bajić, I.V., Lin, W., Tian, Y.: Collaborative intelligence: Challenges and opportunities. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8493–8497. IEEE (2021)
3. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: ICLR (2017)
4. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. In: ICLR (2018)
5. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact: Real-time instance segmentation. In: ICCV. pp. 9157–9166 (2019)
6. Bross, B., Wang, Y.K., Ye, Y., Liu, S., Chen, J., Sullivan, G.J., Ohm, J.R.: Overview of the versatile video coding (vvc) standard and its applications. TCSVT (2021)
7. Cai, Q., Chen, Z., Wu, D., Liu, S., Li, X.: A novel video coding strategy in hevc for object detection. TCSVT (2021)
8. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882 (2020)
9. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI **40**(4), 834–848 (2017)
10. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV. pp. 801–818 (2018)
11. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607. PMLR (2020)
12. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
13. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR. pp. 15750–15758 (2021)
14. Chen, Z., He, T., Jin, X., Wu, F.: Learning for video compression. IEEE Transactions on Circuits and Systems for Video Technology **30**(2), 566–576 (2019)
15. Chen, Z., Duan, L.Y., Wang, S., Lin, W., Kot, A.C.: Data representation in hybrid coding framework for feature maps compression. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 3094–3098. IEEE (2020)
16. Chen, Z., Fan, K., Wang, S., Duan, L.Y., Lin, W., Kot, A.: Lossy intermediate deep learning feature compression and evaluation. In: ACM MM. pp. 2414–2422 (2019)
17. Chen, Z., Fan, K., Wang, S., Duan, L., Lin, W., Kot, A.C.: Toward intelligent sensing: Intermediate deep feature compression. TIP **29**, 2230–2243 (2019)
18. Chen, Z., Lin, W., Wang, S., Duan, L., Kot, A.C.: Intermediate deep feature compression: the next battlefield of intelligent sensing. arXiv preprint arXiv:1809.06196 (2018)
19. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: CVPR. pp. 12475–12485 (2020)

20. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: CVPR. pp. 7939–7948 (2020)
21. Choi, H., Bajic, I.V.: High efficiency compression for object detection. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1792–1796. IEEE (2018)
22. Codevilla, F., Simard, J.G., Goroshin, R., Pal, C.: Learned image compression for machine perception. arXiv preprint arXiv:2111.02249 (2021)
23. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation (2020)
24. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. pp. 3213–3223 (2016)
25. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. Ieee (2009)
26. Duan, L., Liu, J., Yang, W., Huang, T., Gao, W.: Video coding for machines: A paradigm of collaborative compression and intelligent analytics. TIP **29**, 8680–8695 (2020)
27. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**(2), 303–338 (2010)
28. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)
29. Guo, Z., Zhang, Z., Feng, R., Chen, Z.: Causal contextual prediction for learned image compression. IEEE Transactions on Circuits and Systems for Video Technology **32**(4), 2329–2341 (2021)
30. Guo, Z., Zhang, Z., Feng, R., Chen, Z.: Soft then hard: Rethinking the quantization in neural image compression. In: International Conference on Machine Learning. pp. 3920–3929. PMLR (2021)
31. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. pp. 9729–9738 (2020)
32. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV. pp. 2961–2969 (2017)
33. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
34. He, T., Sun, S., Guo, Z., Chen, Z.: Beyond coding: Detection-driven image compression with semantically structured bit-stream. In: 2019 Picture Coding Symposium (PCS). pp. 1–5. IEEE (2019)
35. Hu, Y., Yang, S., Yang, W., Duan, L.Y., Liu, J.: Towards coding for human and machine vision: A scalable image coding approach. In: 2020 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2020)
36. Huang, Z., Jia, C., Wang, S., Ma, S.: Visual analysis motivated rate-distortion model for image coding. In: 2021 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2021)
37. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. TPAMI (2020)
38. Johnston, N., Vincent, D., Minnen, D., Covell, M., Singh, S., Chinen, T., Hwang, S.J., Shor, J., Toderici, G.: Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In: CVPR. pp. 4385–4393 (2018)
39. Kim, J.H., Heo, B., Lee, J.S.: Joint global and local hierarchical priors for learned image compression. arXiv preprint arXiv:2112.04487 (2021)

40. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: CVPR. pp. 9404–9413 (2019)
41. Le, N., Zhang, H., Cricri, F., Ghaznavi-Youvalari, R., Rahtu, E.: Image coding for machines: An end-to-end learned approach. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1590–1594. IEEE (2021)
42. Li, M., Zuo, W., Gu, S., You, J., Zhang, D.: Learning content-weighted deep image compression. TPAMI (2020)
43. Li, M., Zuo, W., Gu, S., Zhao, D., Zhang, D.: Learning convolutional networks for content-weighted image compression. In: CVPR. pp. 3214–3223 (2018)
44. Li, X., Shi, J., Chen, Z.: Task-driven semantic coding via reinforcement learning. arXiv preprint arXiv:2106.03511 (2021)
45. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. pp. 2117–2125 (2017)
46. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755. Springer (2014)
47. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR. pp. 8759–8768 (2018)
48. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. pp. 3431–3440 (2015)
49. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
50. Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Van Gool, L.: Conditional probability models for deep image compression. In: CVPR. pp. 4394–4402 (2018)
51. Mentzer, F., Toderici, G.D., Tschannen, M., Agustsson, E.: High-fidelity generative image compression. NeurIPS **33**, 11913–11924 (2020)
52. Minnen, D., Ballé, J., Toderici, G.: Joint autoregressive and hierarchical priors for learned image compression. In: NeurIPS (2018)
53. Minnen, D., Singh, S.: Channel-wise autoregressive entropy models for learned image compression. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 3339–3343. IEEE (2020)
54. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV. pp. 483–499. Springer (2016)
55. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV. pp. 69–84. Springer (2016)
56. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
57. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR. pp. 2536–2544 (2016)
58. Rabbani, M., Joshi, R.: An overview of the jpeg 2000 still image compression standard. Signal processing: Image communication **17**(1), 3–48 (2002)
59. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
60. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: CVPR. pp. 7263–7271 (2017)
61. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. NeurIPS **28**, 91–99 (2015)
62. Singh, S., Abu-El-Haija, S., Johnston, N., Ballé, J., Shrivastava, A., Toderici, G.: End-to-end learning of compressible features. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 3349–3353. IEEE (2020)

63. Song, M., Choi, J., Han, B.: Variable-rate deep image compression through spatially-adaptive feature transform. In: ICCV. pp. 2380–2389 (2021)
64. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (hevc) standard. TCSVT **22**(12), 1649–1668 (2012)
65. Sun, S., He, T., Chen, Z.: Semantic structured image coding framework for multiple intelligent applications. TCSVT (2020)
66. Toderici, G., O'Malley, S.M., Hwang, S.J., Vincent, D., Minnen, D., Baluja, S., Covell, M., Sukthankar, R.: Variable rate image compression with recurrent neural networks. arXiv preprint arXiv:1511.06085 (2015)
67. Wallace, G.K.: The jpeg still picture compression standard. IEEE transactions on consumer electronics **38**(1), xviii–xxxiv (1992)
68. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the h. 264/avc video coding standard. TCSVT **13**(7), 560–576 (2003)
69. Wu, Y., Li, X., Zhang, Z., Jin, X., Chen, Z.: Learned block-based hybrid image compression. IEEE Transactions on Circuits and Systems for Video Technology (2021)
70. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)
71. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR. pp. 3733–3742 (2018)
72. Xia, S., Liang, K., Yang, W., Duan, L.Y., Liu, J.: An emerging coding paradigm vcm: A scalable coding approach beyond feature and signal. In: 2020 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2020)
73. Yang, Y., Bamler, R., Mandt, S.: Improving inference for neural image compression. vol. 33, pp. 573–584 (2020)
74. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV. pp. 649–666. Springer (2016)
75. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR. pp. 2881–2890 (2017)