Residual based hierarchical feature compression for multi-task machine vision

Chaoran Chen[1], Mai Xu[1], Shengxi Li[1]*, Tie Liu[1], Minglang Qiao[1], Zhuoyi Lv[2],
[1] School of Electronic and Information Engineering, Beihang University, Beijing, China
[2] VIVO Mobile Communication Company Limited, Beijing, China

***Abstract*—With the remarkable success of deep learning, image/video coding for machines (VCM) has been playing an important role in facilitating intelligent vision tasks. However, the existing VCM methods suffer from either sub-optimality of using image compression standards, or generalisation issues of learning-based methods. To address these issues, this paper proposes a residual-based hierarchical feature compression (RHFC) method to achieve optimal and universal feature compression for object detection and segmentation. More specifically, we first analyse the redundancy that exists in features at multiple scales, by finding that large-scale features are surprisingly less important to the vision tasks. Thus, we propose a pair of compression and enhancement networks to extract the very basic cues from the large-scale features, which are then compressed by the VVC codec. To compensate the inevitable detail loss, we further propose the hierarchical framework to compress the residuals between the reconstructed and original features, such that the performances can be significantly improved at low bit-rate cost. Experimental results have verified our superior performances, against both the state-of-the-art learning-based and standard feature compression methods. Our RHFC method also generalises well to other scenarios without the need of any further fine-tuning.**

***Index Terms*—Feature compression, image compression, multi-task learning**

I. Introduction

The rapid evolution of multimedia services leads to an explosive growth of image/video data volume, highlighting the emergence of the Big Data era. Overwhelming amount of high-quality images/videos have significantly facilitated people daily life and activities, meanwhile though, posing tough challenges towards limited bandwidth and storage resources. Addressing this challenge thus calls for efficient image/video compression methods to maximally save bit-rates and reduce data volume. Consequently, the past decades have witnessed consistent improvements on the compression efficiency, mainly from standard image/video compression methods, including JPEG, JPEG2000 and H.261-H.266 [1] standards. Most recently, image/video compression also benefits from deep learning techniques [2], [3], by an end-to-end optimisation on important modules such as non-linear transform, quantisation and entropy models, which has improved human-preferred reconstruction given limited bit-rates.

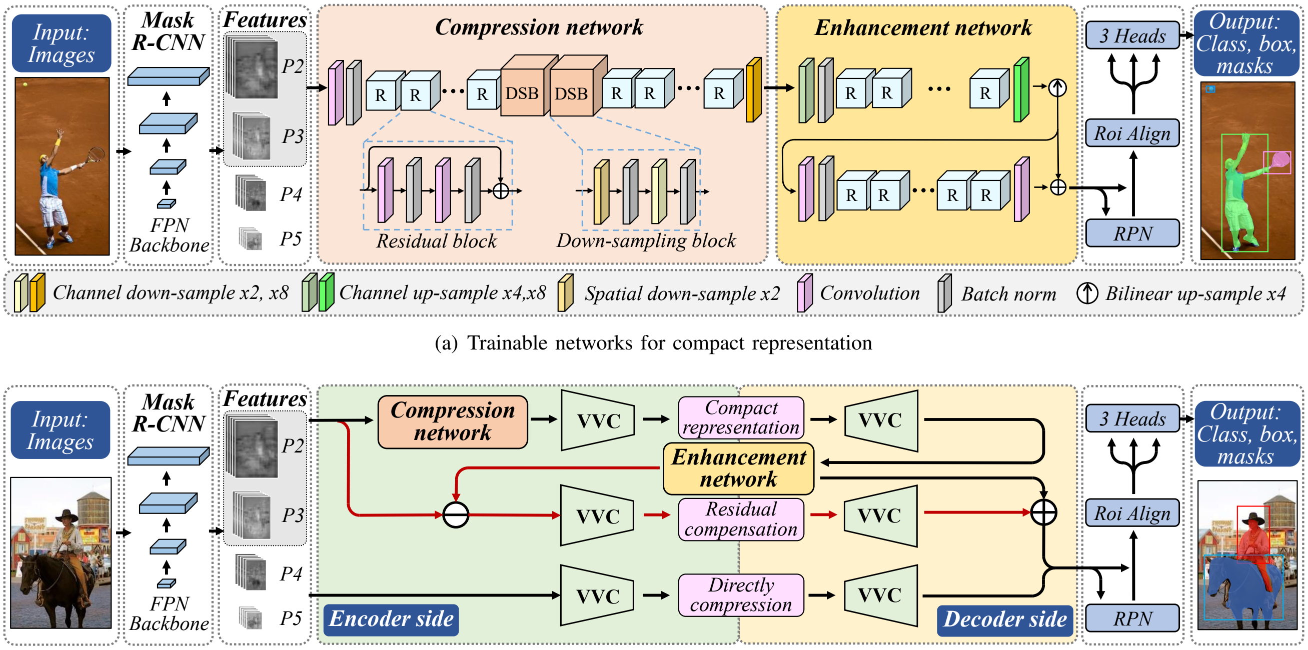
Deep learning techniques have also revolutionised the way we use artificial intelligence (AI) aided and even AI automated machines to accomplish extensive vision tasks, of which the success is indispensable with the Big Data. Given limited transmission and storage resources, it therefore has been an urgent need to seek compression methods that are machine-preferred, instead of human-preferred, so as to reduce the data volume for machine vision tasks such as detection in surveillance, segmentation in anomaly alarming, to name but a few. Catering for machine vision tasks, there are works that insist on compressing images, by either adding machine vision losses in a learnt framework [5], or enhancing image regions that are most important for machine vision tasks during compression [6]–[8]. However, although targeting at machine vision, the above methods have to reconstruct images before completing vision tasks, which may incur both compression redundancy and increased computational complexity.

On the other hand, compressing internal features within machine vision tasks can enjoy improved efficiency. Most recently, the Moving Picture Experts Group (MPEG) has organised the video coding for machines (VCM) group, whereby the preliminary anchor was established to compress internal features from the Mask R-CNN architecture [4] for multi-task detection and segmentation [9], [10]. Such an anchor basically concatenated features to construct large grey-scale images, which are then compressed by the latest VVC/H.266 all-intra codec. To fully make use of the state-of-the-art codec, several proposals attempt to remove the spatial redundancy from the perspective of super-resolution [11], or reorder channels to formalise pseudo-video sequences to reduce channel-wise redundancy [12]. Few works proposed to optimise the compression in an end-to-end learnt manner [13], [14]. Although being expected to improve efficiency, the learning-based method suffers from restricted generalisation capability, which requires repeated training on different bit-rates and datasets. Compared to images, features are with increased float precision and with larger uncompressed sizes due to the existence of channels, which remarkably increases the convergence time and threatens the stability at each training process. Moreover, we noticed several works that tried to combine the learnt and standard methods for image compression [15], [16]. However, they fail to handle the feature compression, whereby the learnt compression can still incur unaffordable bit-rates.

In this paper, we propose a residual-based hierarchical

Corresponding author: Shengxi Li (ShengxiLi2014@gmail.com).

This work was supported by Beijing Natural Science Foundation under Grant JQ20020 and by NSFC under Grants 62250001, 62231002 and 62206011.



(a) Trainable networks for compact representation

(b) Hierarchical framework for compressing features

Fig. 1. The overall architecture of the proposed RHFC method, to compress features from the Mask R-CNN architecture [4], for both object detection and segmentation. FPN denotes feature pyramid network, and RPN is the region proposal network in the Mask R-CNN architecture. (a): The compression and enhancement networks are first trained in an end-to-end manner to obtain compact representations of P_2 and P_3 features. Note that the network modules in the Mask R-CNN are fixed during training for fair comparisons. (b): The hierarchical framework for compressing the compact representations of the P_2 and P_3 features, whereas P_4 and P_5 features are encoded by an extra VVC/H.266 all-intra codec. Note that the compression part of residuals is highlighted by red arrows.

feature compression (RHFC) framework, with the ultimate goal of befitting both optimality (of learning methods) and generalisation (of compression standards). More specifically, we first analyse the redundancy of multi-scale features, which are the workhorse in multi-task for object detection and segmentation. Our analyse points out that the large-scale features possess higher redundancy but are less important to vision tasks, compare to the small-scale features. Then, features at multiple scales are treated by different compression strategies. For large-scale features, we propose the compression and enhancement networks to obtain compact representations by an end-to-end optimisation. Since the pair of compression and enhancement networks aims to extract key cues from the original features, the compact representation generalises well to other dataset for this multi-task scenario, even without the need of fine-tuning. More importantly, the compact representation aims to remove both spatial and channel-wise redundancy, which significantly reduces the feature data volume for compression. The so well-trained compact representation is then compressed by the latest VVC/H.266 all-intra codec. The compression, however, can inevitably lead to distorted compact representations, which mislead the enhancement network to reconstruct details of the original features. This motivates us to compress the residual again by VVC/H.266 all-intra codec to recover the missing details. This constitutes the hierarchical structure of our RHFC method. Moreover, we develop an additional VVC/H.266 all-intra codec to straightforwardly compress the small-scale features, such that all features can be recovered at the decoder side.

Besides, we noticed a recent work named multi-scale feature compression (MSFC) [13] that aims to compress features from the Mask R-CNN architecture, which fuses the multi-scale features before feeding to the compression network. We may need to point out that our work compresses feature per each scale with detailed analysis, and proposes the novel hierarchical framework based on feature residuals, which successfully benefits the learnt optimality and standard generalisation; those new modules distinguish our RHFC method from the MSFC method. In our experiments, when trained solely on COCO dataset [17], our RHFC method achieves the state-of-the-art performances on compressing both features of the COCO and OpenImage test images, saving 84.06% (88.46%) and 91.58% (88.72%) Bjontegaard delta (BD) rates for objection detection (and segmentation), respectively. Our contributions are mainly three-fold:

- We analyse the redundancy of features at multiple scales regarding machine vision tasks, which reasonably guides scale-specific feature compression strategies.
- We propose to extract compact representations of features by the compression and enhancement networks, which removes spatial and channel-wise redundancy, and enables generalisation to other scenarios.
- We propose the hierarchical framework to sequentially compress the crucial cues of the compact representations and remaining details from the residuals, which benefits both the learnt and standard compression.

II. Proposed Method

A. Overall architecture

Fig. 1 illustrates our RHFC method by highlighting the key components, whereby Fig. 1-(a) presents the training stage and Fig. 1-(b) designates our hierarchical testing procedure for compression. Following the common split point that has been suggested by MPEG VCM group [9], [10], our method is designed for compressing the multi-scale pyramid features $\{P_2, P_3, P_4, P_5\}$ from the Mask R-CNN backbone [4]. Generally speaking, regarding the Mask R-CNN architecture specified by the MPEG VCM group, we first analyse the importance of features at multiple scales, and conclude that large-scale features are more redundant compared to the small-scale ones, resulting in the classified treatment adopted by our RHFC method. More importantly, since crucial cues within features that determine the final prediction are typically sparse in machine vision tasks (e.g., object detection and instance segmentation), our RHFC aims to learn the compact representation of those cues at a cost of extremely low bit-rates; this is achieved by an end-to-end training style between the compression and enhancement networks (as shown in Fig. 1-(a)), targeting at optimising multi-task losses for object detection and instance segmentation. Such optimality can benefit the performance of our RHFC method at low bit-rates. To further improve our performance at high bit-rates, the hierarchical module is proposed to compensate the missing details, by compressing the residuals, of which the pipeline has been highlighted by red arrows in Fig. 1-(b). In other words, if we lost everything by the compression and enhancement networks, the residuals now equal to the original features, such that compressing residuals is equivalent to compressing the original features, i.e., the pipeline that has been accepted as the feature anchor by MPEG VCM group. Therefore, the proposed hierarchical structure ensures that our method can at least recover the feature anchor performances, whilst being potential to boost the performances by the proposed hierarchical structure. This way, the performance of our RHFC method can be significantly improved at both low and high bit-rates.

B. Analysis on multi-scale features

Before proceeding further to our RHFC method, we first analyse the redundancy that exists in multi-scale features to be compressed. Although the feature pyramid network (FPN) has been widely applied to detect and segment objects at different scales, the resulting multi-scale features are typically with a large amount of uncompressed data volume, even much larger than the original images. This is not surprising as it targets at a different goal from compression. It is thus necessary to inspect the features at different scales so as to find out possible redundancy that leaves for the compression.

Without loss of generality, we performed preliminary experiments upon the detection task, based on the Faster R-CNN X101-FPN architecture [18] and $5k$ test images from the OpenImage dataset [9], [10], [19], [20]. Note that the Mask R-CNN architecture [4] can share the same backbone with the Faster R-CNN. Indeed, as reported in Table I, the total size of P_2-P_5 features is much larger that that of the image, which means that redundancy exists both within features and across multiple scales. More importantly, it can also be found from Table I that P_2 and P_3 occupy dominant uncompressed sizes, whereas removing them by setting to 0 witnessed merely slight decrease on the mean average precision at threshold 0.5 (mAP@0.5). In contrast, for the P_4 and P_5 features, they play central roles in ensuring mAP performances, whilst occupying the least uncompressed sizes. This phenomenon promotes a different treatment on $\{P_2, P_3\}$ features and $\{P_4, P_5\}$ features in compression. Thus, in the proposed RHFC method, we only compress P_2 and P_3 features by our compression and enhancement networks to extract key cues, whilst leaving P_4 and P_5 features to be directly encoded by the VVC/H.266 all-intra codec.

TABLE I

Details of P_2, P_3, P_4, P_5 features from the Faster R-CNN X101-FPN backbone [18]. The mean average precision at threshold 0.5 (mAP@0.5) is reported by setting the corresponding feature to zero, for detection on $5k$ OpenImage test images [9], [10], [19], [20]. Note that the mAP@0.5 is 79.23% for non-zero P_2, P_3, P_4, P_5 features, i.e., the raw features.

Layer	Size ratio	Size (C, H, W)	mAP@0.5
Input image	12	$(3, h, w)$	–
P_2	64	$(256, {}^{h}/_{4}, {}^{h}/_{4})$	74.22%
P_3	16	$(256, {}^{h}/_{8}, {}^{h}/_{8})$	76.29%
P_4	4	$(256, {}^{h}/_{16}, {}^{h}/_{16})$	18.38%
P_5	1	$(256, {}^{h}/_{32}, {}^{h}/_{32})$	30.50%

C. Compression network

Since P_2 and P_3 are the most redundant features as analysed above, we propose to reduce both spatial sizes and channel numbers of the original P_2 and P_3 features. More specifically, the P_2 and P_3 features are first fed to one reflection padding of size 3 and one convolution layer with kernel size equal to 7, denoted by RConv for short. The RConv operation is followed by a batch normalisation layer and 6 residual blocks. By far, the feature sizes remain unchanged. Then, two down-sampling blocks (DSBs) are employed to reduce both spatial sizes and channel numbers of features, each of which down-samples by a factor of 2. Afterwards, 6 residual blocks are employed, followed by a final reflection padding and convolution layer of kernel size 7, after which the spatial sizes remain unchanged. However, the final convolution layer down-samples the channel number by a factor of 8, and we denote this channel down-sampling RConv operation as DRConv. The overall main pipeline of our compression network can be formulated as

$$\mathbf{f}_c = \mathrm{DRConv}_{\downarrow 8} \circ \mathrm{ResB}_{\times 6} \circ \mathrm{DSB}_{\times 2} \circ \mathrm{ResB}_{\times 6} \circ \mathrm{BN} \circ \mathrm{RConv}(\mathbf{f}), \tag{1}$$

where $\circ$ represents compositions of networks, $\mathbf{f}$ denotes the input feature, either P_2 or P_3, and $\mathbf{f}_c$ is the corresponding output compact representation. In (1), $\mathrm{ResB}(\cdot)$ is for residual blocks and $\mathrm{BN}(\cdot)$ for batch normalisation. Consequently, the proposed compression network down-samples the spatial sizes by a factor of 4 and the channel numbers by 32, which results into the compact representation with extremely small sizes for the subsequent compression.

D. Enhancement network

The key cues extracted from the compression network have to be able to recover the vision task performances of the original features. Our enhancement network thus aims to restore the original P_2 and P_3 features, from the compact representations $\mathbf{f}_c$. More specifically, the number of channels of the compact representation is up-sampled by a factor of 4, under the operation of the reflection padding and convolution layer. Again, this operation does not change the spatial size of input, and is denoted by URConv. Then, 6 residual blocks are employed and maintain the sizes unchanged, followed by another URConv by a factor of 8. This way, the original 256 channels are recovered. For the spatial sizes, we then bi-linearly up-sample the features to the original sizes of P_2 and P_3 features, respectively. At this stage, the recovered features are with the same sizes of the P_2 and P_3 features, which can be written by

$$\mathbf{f}_{\text{up}} = \text{SUp}_{\uparrow 4} \circ \text{URConv}_{\uparrow 8} \circ \text{ResB}_{\times 6} \circ \text{BN} \circ \text{URConv}_{\uparrow 4}(\mathbf{f}_c), \tag{2}$$

where $\text{SUp}_{\uparrow 4}(\cdot)$ denotes spatial up-sampling by the bi-linear operator; recall that $\text{URConv}_{\uparrow 4}(\cdot)$ and $\text{URConv}_{\uparrow 8}(\cdot)$ are channel up-sampling with spatially unchanged operations, consisting of reflection padding and convolution by the factors of 4 and 8, respectively.

Then, the size-recovered features $\mathbf{f}_{\text{up}}$ are then sequentially processed by one RConv, batch normalisation, 9 residual blocks and a final Rconv operation, with the goal to predict the residual between the restored and original P_2 (or P_3) feature. The prediction loss can be defined by the mean squared error (MSE) as follows,

$$\mathcal{L}_{\text{mse}} = ||\hat{\mathbf{f}} - \mathbf{f}||_2^2, \tag{3}$$

where $\hat{\mathbf{f}}$ denotes the restored feature map for either P_2 and P_3 features, and is obtained by

$$\hat{\mathbf{f}} = \text{RConv} \circ \text{ResB}_{\times 9} \circ \text{BN} \circ \text{RConv}(\mathbf{f}_{\text{up}}) + \mathbf{f}_{\text{up}}. \tag{4}$$

The output of the enhancement network is fed into the region proposal network (RPN) for region of interest (ROI) alignment, such that the multiple tasks can be accomplished by their corresponding heads. We therefore train the compression and enhancement networks by the following multi-task loss in an end-to-end manner:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{mask}} + \lambda \mathcal{L}_{\text{mse}}, \tag{5}$$

where $\mathcal{L}_{\text{cls}}$, $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{mask}}$ represent the classification accuracy, bounding box regression and binary mask cross-entropy losses [4]; λ is the hyper-parameter to balance the reconstruction loss $\mathcal{L}_{\text{mse}}$ of features. Note that although fine-tuning may witness further improvements, the Mask R-CNN architecture is fixed during training for fair comparisons [9], [10], [13], [20]. Indeed, it only provides gradients for training our compression and enhancement networks.

E. Residual-based hierarchical feature compression (RHFC)

After training compression and enhancement networks, we apply them to compress P_2 and P_3 features in our test phase. To further reduce the data volume of compact representations of P_2 and P_3 features, we employ VVC/H.266 all-intra codec to compress those compact representations $\mathbf{f}_c$. However, involving the VVC compression in between can lead to the mismatch between the well-trained compression and enhancement networks. Thus, we propose the hierarchical framework to compress the residuals between the reconstructed and original features, such that the reconstructed features from the VVC compressed representation are compensated to retain details of the original features. Since the reconstructed features from the enhancement network are regularised by $\mathcal{L}_{\text{mse}}$ during training, the residuals are minimised, such that the additional bit-rate cost for compressing the residuals can be maximally reduced. On the other hand, for the P_4 and P_5 features, they are compressed by an extra VVC/H.266 all-intra codec. Note that following [9], [10], all the features are re-scaled and concatenated across spatial sizes, to constitute large grey-scale images for VVC compression.

III. Experiments

Experimental settings: We implemented our proposed RHFC method based on the *Facebook Detectron2* platform, which is the widely applied Mask R-CNN architecture. The pre-trained weight was chosen as the *X-101-FPN* configuration, for fair comparisons to the existing feature compression methods. We comprehensively compared our method against the state-of-the-art learning-based and standard compression methods. More specifically, for the learning-based methods, MSFC [13], Gao [14] and Zhu [7] were employed for comparisons with their default settings, which benefit from the optimality of the end-to-end learnt style. On the other hand, the compared feature [9] and image [20] anchors are the standard compression methods that enjoy the generalisation capability, as specified by the MPEG VCM group. Please note that for the standard compression methods, we compared our method against the same H.266/VVC platform, namely, VTM-12.0, for fair evaluations. Instead of resizing the short edge of images to 640 800 during data pre-processing, we empirically resized to 416 when compressing the COCO dataset and 576 for the OpenImage dataset. During the training phase, we trained our compression and enhancement networks in an end-to-end manner, upon COCO training set that contains $118k$ images with 80 categories. Meanwhile, we fixed the weights within the Mask R-CNN architecture as specified [9], [10]. During our test phase, we employed QP=22 to compress the compact representation, so as to ensure high quality of important cues. The QPs for compressing P_4 and P_5 features were set to $\{\text{QP}_i\}_{i=1}^5 = \{35, 37, 39, 41, 43\}$ when compressing $5k$ COCO validation images. Moreover, when compressing the residuals, we used an incremental mapping

TABLE II
RATE-ACCURACY COMPARISONS ON THE COCO VALIDATION SET FOR THE OBJECT DETECTION AND INSTANCE SEGMENTATION.

Feature anchor [9], [10]			Image anchor [20]			Our method		
bpp	o.d.	i.s.	bpp	o.d.	i.s.	bpp	o.d.	i.s.
1.44	52.91	47.34	0.91	51.07	47.89	0.37	53.12	48.31
0.97	50.50	43.87	0.52	50.04	46.85	0.25	52.10	47.39
0.65	46.99	39.29	0.27	47.49	44.40	0.17	51.13	46.58
0.43	42.33	33.17	0.13	42.12	38.93	0.12	50.08	45.23
0.28	35.31	25.07	0.06	31.54	29.00	0.09	49.44	44.24
BD-R	–	–	BD-R	-63.05	-76.42	BD-R	-84.06	-88.46

Note: o.d. denotes mAP@0.5 (%) for the object detection, and i.s. represents mAP@0.5 (%) for the instance segmentation. BD-R is Bjontegaard delta rate (%).

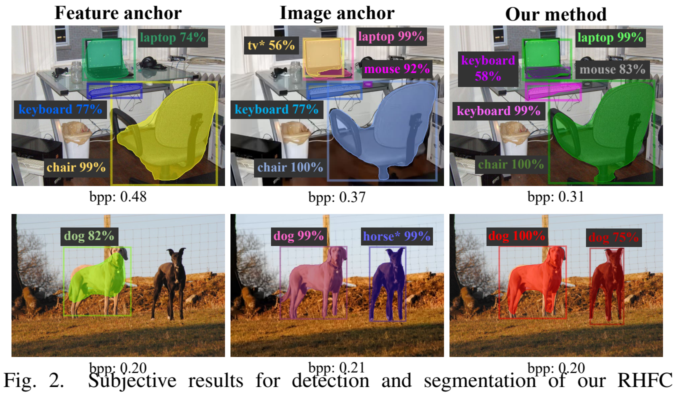


Fig. 2. Subjective results for detection and segmentation of our RHFC method, compared with the feature and image anchors. The first row exhibits the results from the COCO dataset, whereas the second row is from the OpenImage dataset. Note that wrongly detected objects are annotated by the symbol $*$.

as $\overline{\text{QP}}_i = \text{QP}_i + (i-1)$ so as to further save bit-rates on low bit-rates. More importantly, to verify the generalisation of our method, we also compressed features from the standard OpenImage $5k$ test images [9], [10], [19], [20] without any further fine-tuning on the OpenImage dataset [19], whereby the QPs were set to $\{\text{QP}_i\}_{i=1}^{6} = \{35, 37, 39, 41, 43, 45\}$. The bit per pixel (bpp) and mAP@0.5 were employed as the metrics to evaluate the feature compression efficiency. Besides, our code is available at https://github.com/17-Ranger/VCM_DSSLIC.

Evaluations on the COCO dataset: We first evaluated the performance of our RHFC method on the COCO validation set, which contains $5k$ images. The results are reported in Table II. The 5 bit-rates of the feature anchor were obtained by setting QPs to $\{35, 37, 39, 41, 43\}$, whereby those for the image anchor were $\{27, 32, 37, 42, 47\}$. From Table II, it is obvious that our method consistently achieves the best compression performances across all the bit-rates, which witnessed more than 84% BD-rate saving against the feature anchor. In fact, our method also achieved more than 65% BD-rate saving when compared with the image anchor. The subjective results are also provided in the first row of Fig. 2. As can be seen from this figure, our method, with the lowest bpp, achieved the most accurate detection and segmentation.

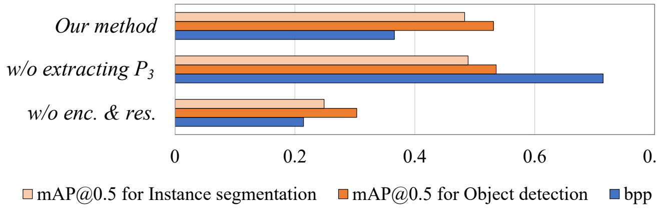


Fig. 3. Rate-accuracy results in ablation study experiments.

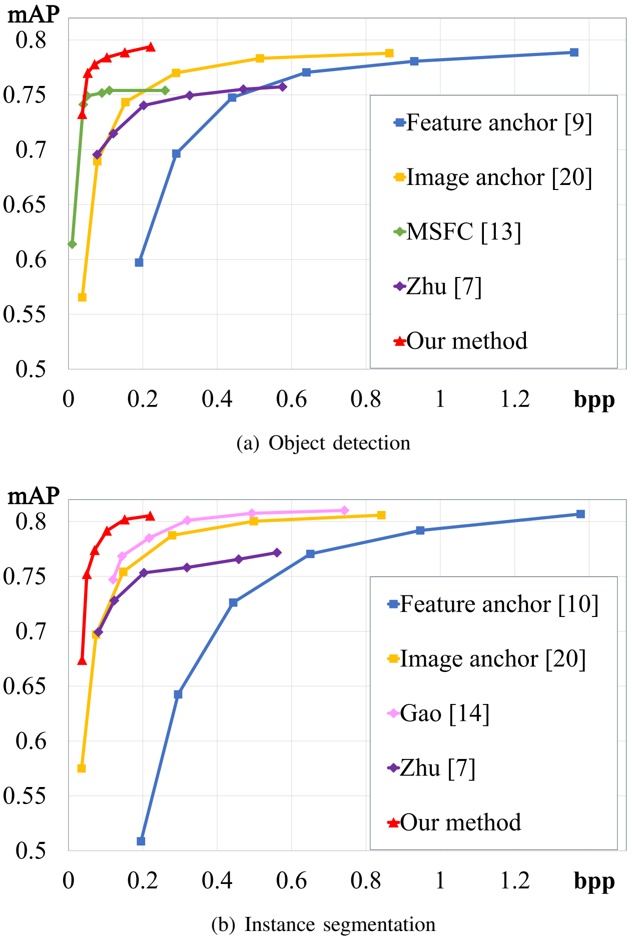
TABLE III
RATE-ACCURACY COMPARISONS ON THE OPENIMAGE $5k$ TEST IMAGES FOR OBJECT DETECTION AND INSTANCE SEGMENTATION.

Object detection									
F. anchor [9]		I. anchor [20]		MSFC [13]		Zhu [7]		Our method	
bpp	mAP	bpp	mAP	bpp	mAP	bpp	mAP	bpp	mAP
1.36	78.87	0.86	78.80	0.26	75.40	0.58	75.73	0.22	79.39
0.93	78.06	0.52	78.33	0.11	75.40	0.47	75.51	0.15	78.87
0.64	77.05	0.29	77.00	0.09	75.17	0.33	74.95	0.10	78.41
0.44	74.75	0.15	74.34	0.05	74.90	0.20	74.04	0.07	77.78
0.29	69.65	0.08	68.96	0.04	74.19	0.12	71.47	0.05	77.00
0.19	59.72	0.04	56.55	0.01	61.40	0.08	69.56	0.04	73.25
BD-R	–	BD-R	-67.48	BD-R	nan	BD-R	-52.66	BD-R	-91.58
Instance segmentation									
F. anchor [10]		I. anchor [20]		Gao [14]		Zhu [7]		Our method	
bpp	mAP	bpp	mAP	bpp	mAP	bpp	mAP	bpp	mAP
1.38	80.68	0.84	80.58	0.74	81.01	0.56	77.17	0.22	80.53
0.95	79.18	0.50	80.02	0.49	80.75	0.46	76.56	0.15	80.19
0.65	77.04	0.28	78.74	0.32	80.11	0.32	75.81	0.10	79.16
0.44	72.61	0.15	75.42	0.22	78.50	0.20	75.33	0.07	77.40
0.30	64.25	0.07	69.70	0.14	76.84	0.12	72.80	0.05	75.21
0.20	50.86	0.04	57.50	0.12	74.71	0.08	69.93	0.04	67.39
BD-R	–	BD-R	-76.31	BD-R	-75.65	BD-R	-65.87	BD-R	-88.72

Note: F.anchor denotes the feature anchor, and I. anchor is the image anchor. BD-R is Bjontegaard delta rate (%) against the feature anchor. mAP denotes mAP@0.5 (%).

Ablations on the COCO dataset: We conducted two ablations to verify the effectiveness of scale-specific compression strategy and the residual-based framework in our RHFC method. To this end, instead of extracting compact representations for P_2 and P_3 features, we chose to only represent P_2, whilst leaving P_3 to be compressed together with P_4 and P_5 features; this is denoted as *w/o extracting* P_3. The resulting compression performance is shown in Fig. 3. Thus, although witnessing slight improvement on accuracy, this variant suffers significant bit-rate cost increase. Moreover, Fig. 3 also illustrates our second ablation, which removed the residual-related components from the enhancement network and the hierarchical residuals during the test phase (denoted by *w/o enc. & res.*). From Fig. 3, *w/o enc. & res.* remarkably distorted the reconstructed features for object detection and instance segmentation, whereby both their mAP values were lower than 31%. Therefore, our method, by extracting key cues for redundant large-scale features and then compressing their residuals hierarchically, enjoys superior VCM performances for the object detection and instance segmentation tasks.

Generalisation on the OpenImage dataset: Since our RHFC method employs the hierarchical framework, we directly applied the RHFC trained on the COCO dataset to compress features from the OpenImage images, without further fine-tuning. The results are listed in Table III, whereby the rate-accuracy curves are plotted in Fig. 4. For the feature and image anchors, the QP values were the same as those when compressing COCO images. Again, we can conclude from this figure that our method enjoys the superior performances against all the comparing methods, including both the learning-based and standard methods. More importantly, our method exhibits the ease of computational complexity because we do



(a) Object detection

(b) Instance segmentation

Fig. 4. Rate-accuracy curves of our method and 4 state-of-the-art comparing methods when compressing OpenImage $5k$ test images, for (a) object detection and (b) instance segmentation.

not need to re-train models when varying bit-rates and even changing datasets. From Table III, our method achieves approximately 90% BD-rate savings for both detection and segmentation tasks, against the standard feature anchor, whereby the subjective results are shown in the second row of Fig. 2.

IV. Conclusion

In this paper, we have proposed the residual-based hierarchical feature compression (RHFC) method, to benefit both from the learning-based and standard compression methods. We first analysed the redundancy in multi-scale features and proposed to remove the spatial and channel-wise redundancy within large-scale features, by constructing the compression and enhancement networks; this obtained the compact representation of features at the extremely low cost of bit-rates. To address the inevitable information loss by compressing the compact representation, we proposed to hierarchically compress the residuals to guarantee the performance at high bit-rates, by the standard VVC codec. The small-scale features were directly compressed by the VVC codec as well. This way, our RHFC can learn to extract compact representation of features to remarkably save bit-rates, whilst allowing for universal high-quality compression on different bit-rates and scenarios. Experimental results have demonstrated that our method can effectively reduce the redundancy in the features, and achieved the state-of-the-art performances on both the object detection and instance segmentation tasks. The superior performances have been consistently highlighted on the OpenImage dataset, even compressed by our RHFC method trained on the COCO dataset, which verified the generalisation of the proposed method.

References

[1] B. Bross, Y. Wang, Y. Ye, S. Liu, J. Chen, G. Sullivan, and J. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE TCSVT*, vol. 31, no. 10, pp. 3736–3764, 2021.

[2] G. Toderici, S. O'Malley, S. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *arXiv preprint arXiv:1511.06085*, 2015.

[3] J. Ballé, V. Laparra, and E. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE ICCV*, 2017, pp. 2961–2969.

[5] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, and E. Rahtu, "Image coding for machines: An end-to-end learned approach," in *IEEE ICASSP*. IEEE, 2021, pp. 1590–1594.

[6] Z. Huang, C. Jia, S. Wang, and S. Ma, "Visual analysis motivated rate-distortion model for image coding," in *IEEE ICME*. IEEE, 2021, pp. 1–6.

[7] B. Zhu, L. Yu, and D. Li, "Deep learning-based compression for machine vision," in *ISO/IEC JTC 1/SC 29/WG 2 m57335*, 2021.

[8] G. Lu, X. Ge, T. Zhong, J. Geng, and Q. Hu, "Preprocessing enhanced image compression for machine vision," *arXiv preprint arXiv:2206.05650*, 2022.

[9] M. Lee, H. Choi, S. Park, M. Kim, S. Oh, D. Sim, Y. Kim, J. Lee, J. Do, and S. Jeong, "EE1.2: P-layer feature map anchor generation for object detection on OpenImagev6 dataset," in *ISO/IEC JTC 1/SC 29/WG 2 m58786*, 2022.

[10] J. Lee, Se Y. Jeong, and Y. Kim, "The test results of compressing P-layer feature maps on the Mask R-CNN network," in *ISO/IEC JTC 1/SC 29/WG 2 m59942*, 2022.

[11] J. Kang, H. Jeong, S. Bae, H. Kim, K. Kim, and S. Jeong, "Feature compression with resize in feature domain," in *ISO/IEC JTC 1/SC 29/WG 2 m59537*, 2022.

[12] Z. Chen, K. Fan, S. Wang, L. Duan, W. Lin, and A. Kot, "Lossy intermediate deep learning feature compression and evaluation," in *ACM-MM*, 2019, pp. 2414–2422.

[13] Z. Zhang, M. Wang, M. Ma, J. Li, and X. Fan, "MSFC: Deep feature compression in multi-task network," in *IEEE ICME*. IEEE, 2021, pp. 1–6.

[14] W. Gao, X. Xu, and S. Liu, "End-to-end learning-based compression for object segmentation," in *ISO/IEC JTC 1/SC 29/WG 2 m58169*, 2021.

[15] M. Akbari, J. Liang, and J. Han, "Dsslic: Deep semantic segmentation-based layered image compression," in *IEEE ICASSP*. IEEE, 2019, pp. 2042–2046.

[16] M. Akbari, J. Liang, J. Han, and C. Tu, "Learned variable-rate image compression with residual divisive normalization," in *IEEE ICME*. IEEE, 2020, pp. 1–6.

[17] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *NIPS*, vol. 28, 2015.

[19] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al., "The open images dataset v4," *IJCV*, vol. 128, no. 7, pp. 1956–1981, 2020.

[20] H. Zhang, N. Le, R. Ghaznavi-Youvalari, F. Cricri, Hamed R. Tavakoli, E. Aksu, M. Hannuksela, T. Ji, K. Misra, P. Cowan, and A. Segall, "VCM anchors on open images dataset," in *ISO/IEC JTC 1/SC 29/WG 2 m57343*, 2021.