IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 January 2023, accepted 23 February 2023, date of publication 22 March 2023, date of current version 7 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3260223

RESEARCH ARTICLE

A Super-Resolution-Based Feature Map Compression for Machine-Oriented Video Coding

JUNG-HEUM KANG[1], MUHAMMAD SALMAN ALI[1], HYE-WON JEONG[1], CHANG-KYUN CHOI[1], YOUNHEE KIM[2], SE YOON JEONG[2], SUNG-HO BAE[1], (Member, IEEE), AND HUI YONG KIM[1], (Member, IEEE)

[1]Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, Republic of Korea
[2]Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea

Corresponding authors: Sung-Ho Bae (shbae@khu.ac.kr) and Hui Yong Kim (hykim.v@khu.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government [Ministry of Science and Information Communication Technology (MSIT)] (Video Coding for Machine) under Grant 2020-0-00011; in part by the IITP grant funded by the Korea Government (MSIT) (Artificial Intelligence Convergence Innovation Human Resources Development, Kyung Hee University) under Grant RS-2022-00155911; in part by the IITP grant funded by the Korea Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068; and in part by the Korea Research Institute for Defense Technology (KRIT) grant funded by the Korea Government [Defense Acquisition Program Administration (DAPA)] under Grant V220023.

ABSTRACT Recently, video and image compression methods using neural networks have received much attention. In MPEG standardization, Video Coding for Machine (VCM) is a newly arising topic which attempts to compress features/images for the purpose of machine vision tasks. Especially, compressing features has advantages in terms of privacy protection and computation off-loading. In this paper, we propose an effective feature compression method equipped with a super-resolution (SR) module for features. Our main motivation comes from the observation that features are somewhat robust to spatial distortions (e.g., AWGN, blur, quantization distortions, coding artifacts), which leads us to integrating an SR module into the compression framework. We also further explore the best training strategy of the proposed method, i.e., finding the best combination of various losses and proper input feature shapes. Our comprehensive experiments show that the proposed method outperforms the baseline in the original VCM anchor scenario on various QP values with Versatile Video Coding (VVC). Specifically, the proposed framework achieved up to 50% BD-rate reduction compared to the conventional P-layer feature map compression method for the object detection task on the OpenImage dataset.

INDEX TERMS Versatile video codec, video coding for machine, feature compression, deep neural network, super resolution.

I. INTRODUCTION

With the recent development of artificial intelligence research, deep neural networks (DNN) outperform the conventional shallow models on various vision tasks such as image classification, object detection, segmentation, and tracking [2], [3], [4], [6], [7], [8]. Likewise, for video/image compression purpose, DNNs have been utilized and shown promising performance. Especially, Video Coding for Machine (VCM), a new rising research field in video compression, has made to effectively compress images and/or features which are to be transmitted for machine vision tasks such as object detection and segmentation [9].

In general, VCM approaches can be categorized into two methods, i.e., feature compression and image compression-based VCM methods. Image compression-based VCM (I-VCM) compresses images using conventional image/video coding tools. On the other hand, feature compression-based VCM (F-VCM) compresses intermediate features in DNNs rather than images. Since features in the early stages of DNNs tend to have lower dimensions than input images, F-VCM may achieve higher compression efficiency.

The associate editor coordinating the review of this manuscript and approving it for publication was Valentina E. Balas.

Moreover, transmitting features can protect privacy due to the difficulty of interpretation of the features by humans. Lastly, F-VCM can enjoy exploiting computational off-loading schemes which can effectively allocate computation burdens to a device (backbone and encoder) and server (decoder and machine vision models) in a collaborative intelligence manner [12], [45].

In this paper, we focus on F-VCM. Specifically, we study an efficient F-VCM scheme based on our preliminary experiments for the distortion-sensitivity characteristics of features. Our experimental findings suggest that features are much less sensitive to various distortions compared to images which lead us to incorporating super-resolution and quantization modules into the proposed F-VCM scheme. Consequently, our comprehensive experiments verify the effectiveness of the proposed F-VCM, showing that the proposed method outperforms the baseline by a large margin in the rate-distortion optimization perspective, verifying the efficiency of the integrated SR module.

Our main contributions are as follows:

- We observed that, in machine vision tasks (e.g., object detection and semantic segmentation), intermediate features in DNNs tends to be more robust than images to various spatial distortions. This observation implies that that proper feature manipulation that introduces sustainable distortions can enhance compression efficiency of the VCM framework.
- Based on the aforementioned observation, we propose a new F-VCM framework that incorporates down and up-scaling modules in the encoder and decoder sides of the F-VCM framework, respectively. Especially for up-sampling, we adopted a super-resolution (SR) network which significantly enhances the coding efficiency.
- We explored optimal conditions for using the SR network within the F-VCM framework. Our results showed that the MSE and SSIM loss combination achieved the best performance among the various loss combinations tested. We also found that, in terms of the input shape of the SR module, using features with multiple channels was a more effective design choice than using single-channel features, based on our experiments.
- Finally, to verify the effectiveness of the proposed method, we perform comprehensive experiments with various SR models on the object detection task. Our experimental results reveal that, compared to the conventional baseline, the proposed framework shows up to 50% higher BD-rate on average over the 6 QP levels. Especially, our method exhibits promising performance with much higher performance gap than the baseline under high QP conditions.

II. RELATED WORKS

A. VIDEO CODING FOR MACHINE

Across the world, digital media content increased exponentially with the growth of content market consumers (e.g., YouTube, Netflix, and Zoom). The size of contents has grown in proportion to their quality (i.e., resolution). Large videos must also be compressed before being transmitted or stored. In response to commercial demands, compression techniques have been developed to reduce bitrates while maintaining visual quality. In the compression pipeline designed exclusively for machine vision tasks, features extracted from DNNs can be replaced with images. In this regard, the MPEG standardization also aims to focus on compressing machine vision task-centric information rather than plain images [42].

Feature compression is an efficient approach of compressing data by employing features as a transmission medium, and has several advantages over image compression. The top and bottom sub-figures in Figure 1 show the conventional image/video coding pipeline and the Feature compression pipeline, each of which encodes, transmits, and decodes DNN images/features. Kang et al. [13] first proposed splitting the computationally expensive DNN into the cloud and edge devices. According to a recent collaborative intelligence study [14], it is possible to efficiently spread compute burdens across cloud and edge devices and reduce energy usage in most circumstances. As mobile devices grow more capable and energy-efficient, performing feature extraction calculations on the front-end mobile device can provide computational power offloading while decreasing energy usage in the back-end data center [11]. Bajić et al. [15] claim that transmitting the features can also prevent privacy issues due to the complexity of feature interpretation by an intercepter. Furthermore, our experimental findings lead to the conclusion that the extracted features have the advantage of being able to tolerate substantial distortions that may occur during data transmission. The aforementioned experiment will be discussed further in Section IV.

B. SINGLE IMAGE SUPER RESOLUTION

Deep neural network (DNN) models have demonstrated impressive performance and strong representational capability on image restoration tasks. SRCNN is the first CNN-based Super Resolution (SR) approach that reconstructs a high-resolution (HR) image from its low-resolution (LR) counterpart using only three convolution layers. Following that, a slew of deeper Single Image Super Resolution (SISR) networks were proposed with the intent of enhancing reconstruction performance [25], [33], [34], [35], [37]. VDSR [25] showed significant performance improvement by using a deeper network with skip connections. Repeated block architectures which are made up of a series of convolutions, activations, skip connections, and other elements, have been popular since the launch of SRResNet [36] and EDSR [37]. Following that, several remarkable approaches, such as MSRN [33], DDBPN [34], and RCAN [35] were proposed, all of which offer promising reconstruction abilities. Feature Super Resolution (FSR) is a domain evolved from SISR research that tries to reconstruct a high-quality discriminative

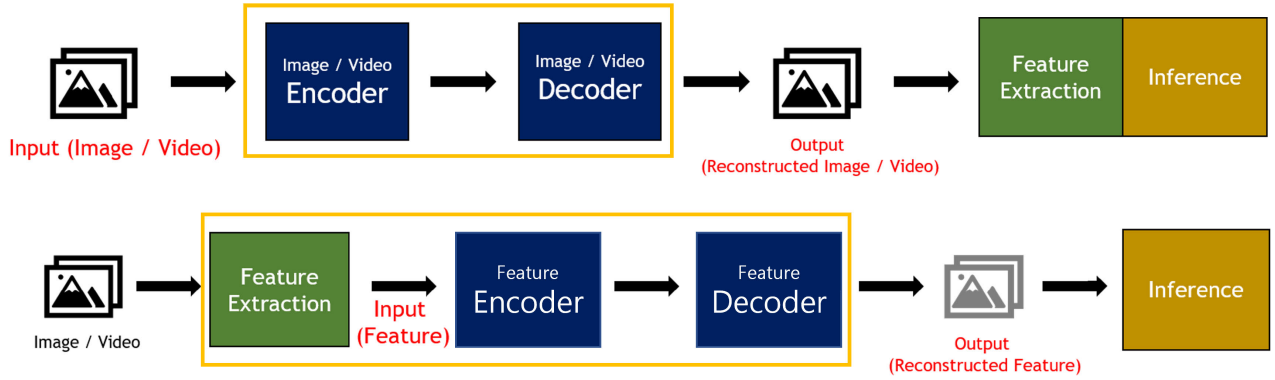

FIGURE 1. Comparison between the generic image compression pipeline and the F-VCM pipeline.

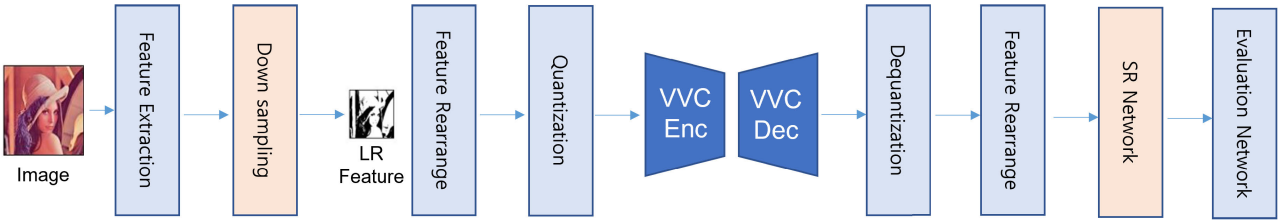

FIGURE 2. Our proposed F-VCM pipeline which incorporate down-scaling, feature rearrangement, quantization, de-quantization, and SR module.

feature from a provided LR ambiguous feature [22], [23]. Preliminary studies use newly designed networks or modules to investigate super resolution of features. However, they focused solely on feature enhancement and did not examine its impact on machine vision applications. To the best of our knowledge, this work is the first to propose using DNN-based SR models for F-VCM in an experimental setting.

III. METHODOLOGY

In this section, we will explain our framework in detail by appending the scaling process into the F-VCM baseline pipeline. To deploy the feature maps in the F-VCM pipeline, additional pre and post-processing steps are required since feature maps are to be compressed using conventional video-codecs. Figure 2 is the overview of our proposed framework. The blue blocks in the pipeline indicate feature compression anchors, whereas the light red blocks represent our proposed down and up-scaling blocks. We follow the predefined anchor setting defined in the 136^{th} MPEG VCM meeting [46].

A. FEATURE EXTRACTION AND DOWN-SCALING

In high-level vision tasks like detection and segmentation, features are extracted using the backbone network. The ResNet-based FPN architecture is used as a backbone network in well-known DNN-based vision task models (e.g., Faster-RCNN [4], Retinanet [5], and Mask-RCNN [6]). In our F-VCM pipeline, we apply compression on P-layer features extracted from the Resnet-based FPN backbone network. The P-layer feature maps are generated through top-down and lateral connection steps using c-layer features from the ResNet backbone intermediate step of feature extraction. We additionally use the bicubic down-sampling to reduce the spatial size of the features, which helps in achieving a reduced bitrate. The downsized features after further pre-processing are then passed through a compressor. After obtaining the restored features from the compressor, with the help of the FSR model, we upscale the features back to their original size and restore a significant portion of the lost high and low-level feature information.

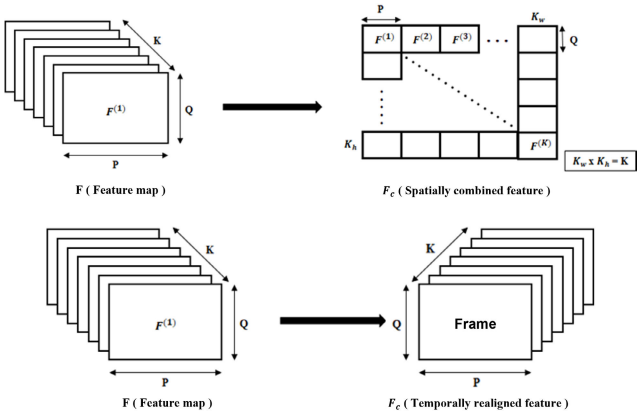

FIGURE 3. The "temporal arrangement" method is depicted in the lower portion of the figure, while the "spatial arrangement" method is shown in the upper portion.

B. FEATURE ARRANGEMENT

Recent research has revealed methods for processing deep features like images using several arrangement schemes that take advantage of the spatial and temporal correlations in the features [26], [45]. Chen et al. [32] evaluated three repacking modules to investigate inter-channel redundancy, and found that the channel tiling approach outperformed the

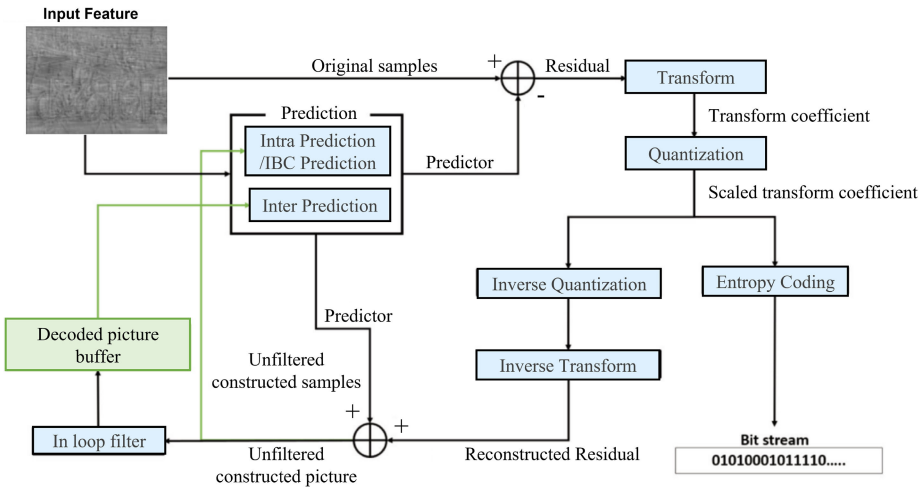

FIGURE 4. The VVC encoder inner architecture, used in the F-VCM pipeline.

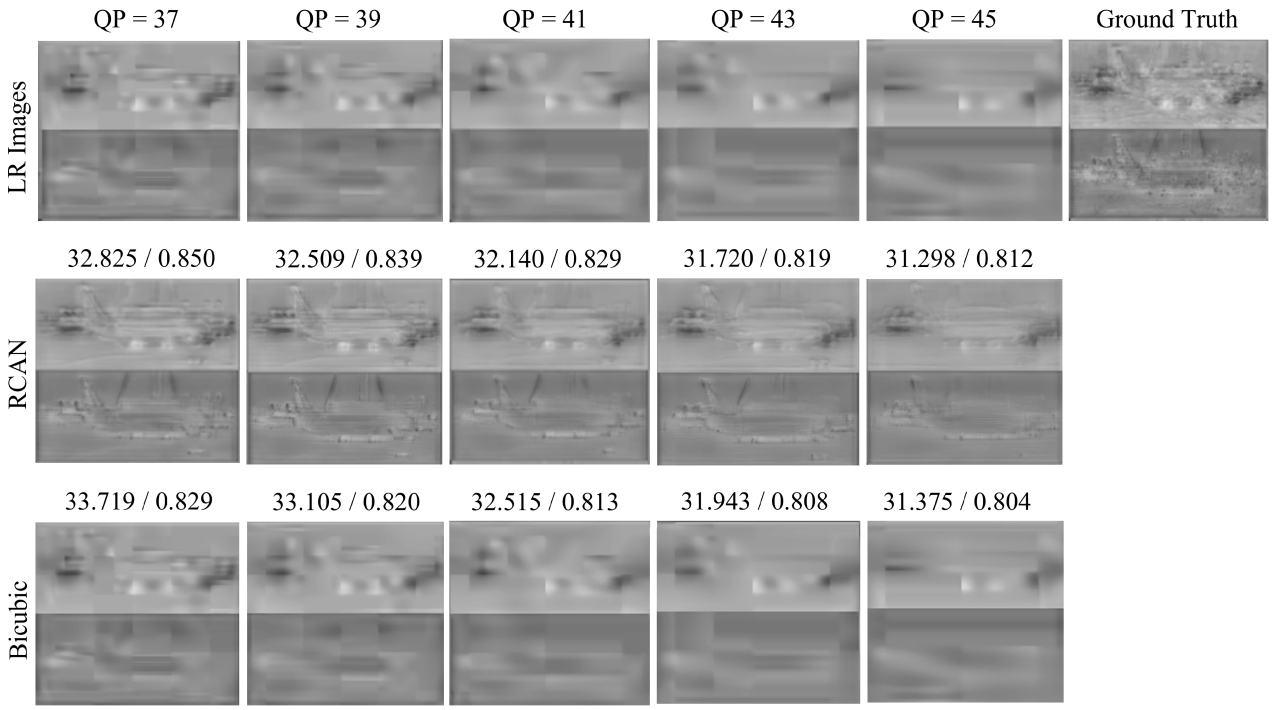

FIGURE 5. Comparison of restoration performance of the proposed training method and conventional method.

other concatenation based approaches on high-level features. Chen et al. [32] also proposed a "temporal arrangement" method that considers the temporal similarity of deep features. The proposed method arranges the deep features temporally and compresses them as a video, as shown in the lower part of Figure 3.

In our F-VCM pipeline, we employ the spatial arrangement method since it is computationally efficient and yields good results compared to the "temporal arrangement" method. It reshapes a three-dimensional feature into a two-dimensional feature, reducing the channel axis by combining the features in width and height directions, as shown in the upper part of Figure 3. Our three-dimensional feature map is transformed into a sizeable two-dimensional feature before it is fed into the compressor using this method.

C. FEATURE QUANTIZATION

The value range of features extracted from the backbone network consists of both positive and negative numbers. The value range of the feature must span from 0 to 255 integer values for compression using a conventional VVC compressor. There are numerous approaches for quantizing a value with a minimal error margin, such as analyzing the value distribution or using k-means clustering. However, preliminary experiments show that the performance loss caused by quantization error in the feature domain is not

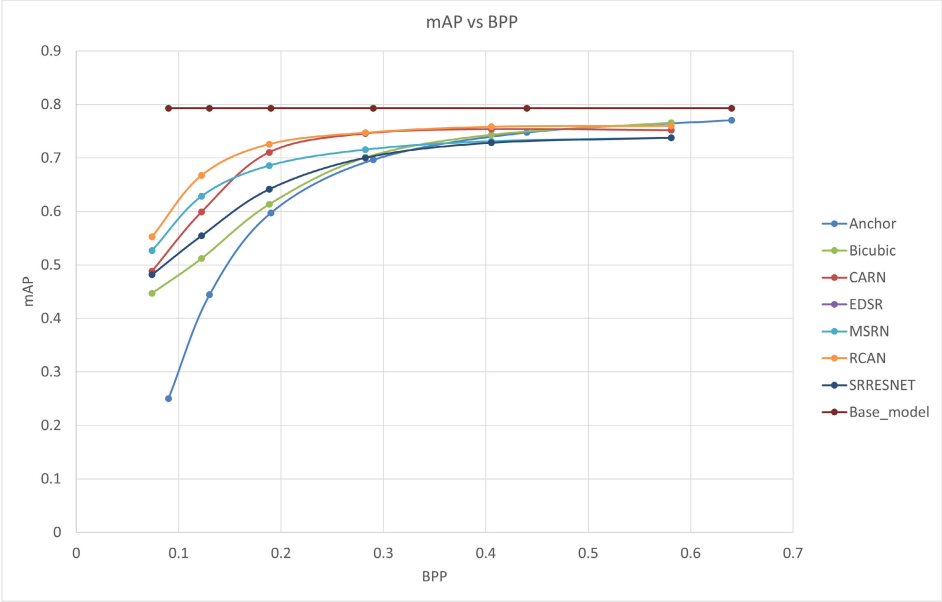

FIGURE 6. RD-performance curve for various FSR models. In comparison to all FSR models, RCAN shows the best performance and achieves a BD-Rate gain of over 50% when compared to the anchor.

TABLE 1. mAP inference result of five different SISR models with various training losses (i.e., L_2 and ($L_2 + L_{SSIM}$)). Since the same feature map is used for up-sampling across all experiments, comparisons are made using the same bpp condition.

Method | Noise Level	QP=35	QP=37	QP=39	QP=41	QP=43	QP=45
Bicubic	0.766	0.7433	0.7008	0.6133	0.5121	0.4467
L_2 Loss for training model						
SRRESNET	0.4656	0.4582	0.4437	0.4437	0.4198	0.3749
EDSR	0.4708	0.4706	0.466	0.4526	0.4382	0.4158
MSRN	0.5175	0.5119	0.4988	0.4788	0.4488	0.4195
CARN	0.6789	0.6681	0.6486	0.6073	0.5402	0.4543
RCAN	0.5725	0.5617	0.543	0.5133	0.476	0.4207
Average	0.5410	0.5341	0.5253	0.5042	0.4685	0.4164
L_2+L_{SSIM} Loss for training model						
SRRESNET	0.7375	0.7282	0.7003	0.6414	0.5545	0.4817
EDSR	0.7478	0.7476	0.7386	0.7150	0.6574	0.5388
MSRN	0.7374	0.731	0.7154	0.6855	0.6284	0.5265
CARN	0.7517	0.7536	0.7456	0.7105	0.5992	0.4881
RCAN	0.7596	0.7582	0.7471	0.7257	0.6675	0.5523
Average	0.7468	0.7437	0.7294	0.6956	0.6214	0.5175

TABLE 2. RD-performance comparison of the anchor method and our proposed method. The QP is adjusted so that it corresponds to the anchor's bpp.

Anchor QP | Our QP	Anchor		Proposed		
	bpp	mAP	bpp	mAP(RCAN)	mAP(Bicubic)
QP41 | QP35	0.44	0.7475	0.5190	0.7596	0.7660
QP43 | QP37	0.29	0.6965	0.3434	0.7582	0.7433
QP45 | QP39	0.19	0.5972	0.2394	0.7470	0.7008
QP47 | QP41	0.13	0.4440	0.1597	0.7256	0.6133
QP49 | QP43	0.09	0.2500	0.1039	0.6675	0.5121
QP49 | QP45	0.09	0.2500	0.0667	0.5523	0.4467

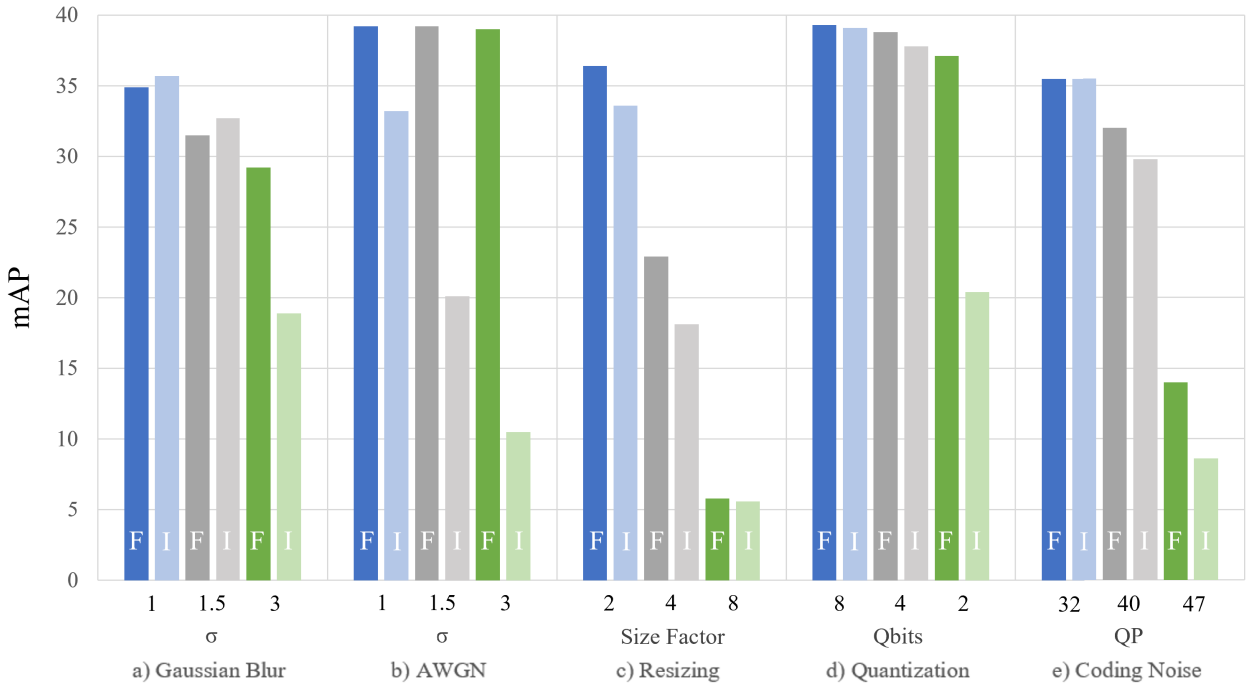

FIGURE 7. Analysis of the effects of different distortions on Images (I) and Features (F).

TABLE 3. RD-performance comparison of the anchor method with our proposed method using single-channel and multi-channel features as input.

QP	Anchor		Channel Type		
	bpp	mAP	bpp	mAP(Single)	mAP(Multi)
QP32	18.335	38.4	5.575	35.6	37.1
QP37	8.117	34.5	2.059	29.4	33.3
QP42	3.277	30.9	0.720	20.2	23.4
QP45	1.823	24.0	0.366	10.2	14.9
QP47	1.212	20.5	0.227	8.8	10.2

TABLE 4. The performance of our proposed method on the object detection task at QP27. The bpp was 6.18 in all cases.

Model	Model Trained on	
	QP27	QP35-QP45
EDSR	0.7878	0.6870
MSRN	0.7641	0.7368
RCAN	0.7870	0.6980

significant (Section IV-F). As a result, to quantize the feature values, we simply utilize the uniform quantization approach with an equal step size in each interval. The uniform quantization is defined as:

$$F_q = Round\left(\frac{(F - V_{min}) \times (2^t - 1)}{V_{max} - V_{min}}\right) \times \left(\frac{V_{max} - V_{min}}{2^t - 1}\right) + V_{min} \quad (1)$$

where F_q denotes the quantized feature value and F denotes the original feature value. In Eq. (1), t is the bit-depth for quantization, and V_{min} and V_{max} are the minimum and maximum value of the whole feature maps.

Dequantization process is performed after the features are received at the decoder side. The uniform dequantization equation is the inverse of the quantization equation, and is expressed as:

$$F_{dq} = F_q \times (V_{max} - V_{min})/(2^t - 1) + V_{min} \quad (2)$$

where F_{dq} and F_q represent the dequantized and quantized feature values, respectively.

D. ENCODER AND DECODER

In our pipeline, we use a VVC-based encoder and decoder for compression. However, for training, we use VVenC [49] and VVdeC [50] as the compressor, since VVenC and VVdeC are faster variants of VVC based on the VVC Test Model (VTM). VVenC/VVdeC removes bottlenecks from the VVC algorithm and increases speed through various optimization and multi-threading-based operations [44]. We utilize the 'faster' preset for training, among several presets presenting a trade-off between running speed and output quality. The compression performance is 2-4 times worse when using the "faster" preset than the "medium" preset. VVenC only supports the YUV420 format, therefore features are transformed to the YUV420 format. We transformed the features to the YUV420 format using FFmpeg 4.2.2 software. As the feature data originally had only one channel, we replicated the data to utilize the three channels required by the YUV420 format. The loop filter is turned off in the experiments as blurring out the regions where the feature window overlap is not necessary. Since VVC is slow and computationally expensive, it is not feasible to deploy VVC during training. As a result, we employ VVenc's 'faster' preset, which is the

TABLE 5. Details of the computational complexity of the SR models compared to Anchor.

	CARN	EDSR	SRRESNET	MSRN	RCAN	Average	Anchor
SR Model Inference Time(sec)	215.7	171.9	194	239.3	176.98	199.57	-
End to End Model Running Time (sec)	2743.1	3200.9	3001.1	3105.4	2148.1	2839.72	842.05
Num of Params(M)	113.6	145.8	137.0	199.9	127.7	144.8	104
SR Model FLOPS (GMac)	948	2204.13	2457.52	4904.38	1273.55	2357.52	-

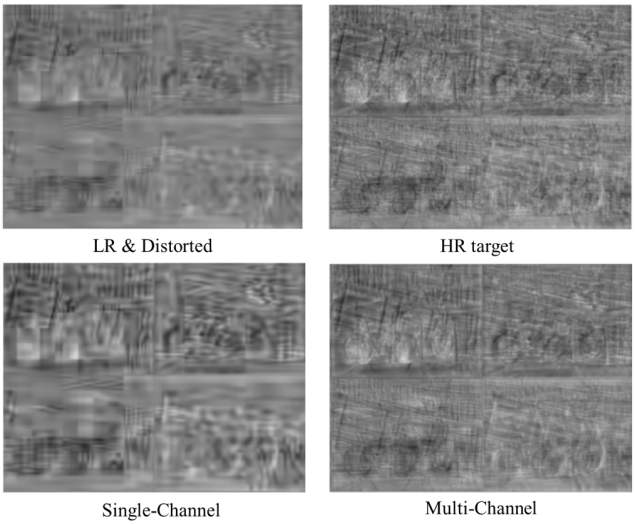

FIGURE 8. The visual quality of the restored multi-channel and single-channel features compared to the HR target.

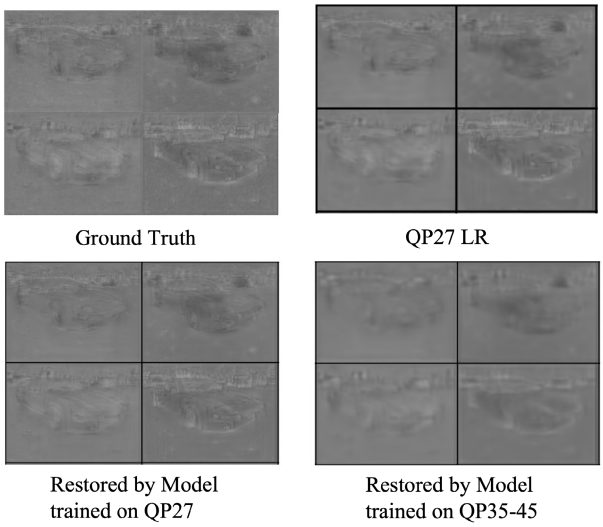

FIGURE 9. Comparison of the restoration performance at QP27 of the model trained on QP27 to the model trained on QP35-QP45, using the ground truth image as a reference. Features are restored using RCAN.

fastest of various presets based on the speed/quality trade-off. We employ VVenC and VVdeC as encoder and decoder during training and use the VVC compressor for testing. Figure 4 depicts the internal working of the VVC. We follow the same anchor settings as defined in the 136^{th} MPEG VCM meeting [46] for fair comparison.

E. FEATURE SUPER RESOLUTION

Using an FSR model, we aim to convert low resolution features to high resolution features while preserving feature quality. This conversion is expressed by:

$$F_{SR} = G(F_{LR}) \tag{3}$$

where $G(\cdot)$ is the FSR model, F_{LR} and F_{HR} refer to low-resolution and high-resolution features, respectively. We deploy the existent state-of-the-art SISR models and replace the input image with features. We chose five well-known SISR models of various computational budget and complexity to test the generality of our proposed pipeline, i.e., SRResNet [36], EDSR [37], MSRN [33], CARN [31], and RCAN [35]). The FSR model's output is then fed into a task evaluation network to measure performance in mAP(Mean Average Precision).

F. OBJECTIVE FUNCTION

In the image restoration task, the objective function employed for training plays a critical role. L2 loss is the most widely used loss function due to its simplicity and good performance. However, it often fails to perform well in the reconstruction of high-quality images as it ignores the interdependence among adjacent pixels and channels. Since our task objective is to reconstruct highly-discriminative features, using L2 loss may not be the ideal choice, as shown in the results in Figure 5. Figure 5 visualizes the L2 loss per pixel for the original LR, RCAN, and Bicubic. Figure 5 shows that RCAN reconstructs the structure of the features in a better fashion than LR and Bicubic reconstructed features, implying that PSNR is less correlated with the representation power of the features. We, therefore, consider other metrics reflecting the visual quality perception of the human visual system (HVS). Accordingly, we adopt the SSIM metric [48], which is commonly used as a quality metric for comparing restoration performance. SSIM is a perception-based metric that treats image degradation as a perceived change in structural information while also taking into account perceptual cues in perceiving quality such as luminance, contrast, and structural similarities. Zhao et al. [38] explored various losses for the denoising task and found that using the combination loss of L1 and MS-SSIM for training preserves the high contrast in restored images.

We investigated various metrics for feature reconstruction with machine vision tasks and found that the weighted sum of SSIM and MSE produced the best results of all the metrics. The loss can be represented as a linear combination L as follows:

$$L = \alpha L_2 + \beta L_{SSIM} \tag{4}$$

where

$$L_2(F) = \sum_{i=1}^{N} |y(i) - \hat{y}(i)|^2 \tag{5}$$

and

$$L_{SSIM}(F) = \sum_{i=1}^{N} 1 - SSIM(y(i), \hat{y}(i)) \tag{6}$$

where i denotes a window in a feature map F with a total of N windows. In Eq. (4), we set α as 10 and β as 1. In Eq. (5) and Eq. (6), y and $\hat{y}$ represent the ground-truth and restored features, respectively. We experimented with several loss function combinations, using perceptual [39] and adversarial losses [40], and discovered that the loss function in Eq. (4) yielded the best results.

G. EVALUATION PROCESS AND METRIC

We evaluated the Mean Average Precision (mAP) performance of the pre-trained Faster-RCNN [4] using 5000 pre-selected images from the OpenImage test dataset [1]. The assessment was carried out using the Detectron2 [41] object detection framework. To verify the feature restoration performance of the FSR models, we used the PSNR and SSIM metrics. Since the pre-trained box predictor generates numerous proposals for each image, the classification model assigns a box proposal score to each prediction box. We calculated the mAP score using the COCO API, considering IOU values ranging from 0.50 to 0.95 for all areas and maxDets values of 100. The IOU score reflects the percentage of overlap between two regions, calculated by dividing the area of overlap by the area of the union. In our case, the IOU score was calculated by comparing the positions of the label and prediction boxes. The maxDets value, which is 100 in our case, determines the maximum number of boxes the model will evaluate.

IV. EXPERIMENTAL RESULTS

A. VCM BENCHMARK STANDARDS

The VCM feature compression standardization meeting selected the benchmark datasets to validate the proposed methodologies. OpenImage [1], CityScape [17] and Tsinghua-Tencent 100k dataset [18] are among the benchmark datasets. Mask-RCNN [6] was selected as an evaluation model for instance segmentation, while Faster-RCNN [4] was selected for object detection.

B. MODELS AND DATASET

We conducted experiments using five distinct SISR models for validating the effectiveness of the proposed F-VCM pipeline. We observed that the complexity of residual connections significantly impacts the proposed method's compression efficiency. As a result, we categorize SISR networks into two types based on the intricacy of residual connections: i) simple models (SRResNet [36], EDSR [37]); and ii) densely connected models (MSRN [33], CARN [31], and RCAN [35]). We employ a single model to handle all the QP distortion levels, similar to various image denoising models that use a single model to handle a variety of unknown noise levels [27], [28], [29]. We partially use the COCO dataset [16] by selecting 500 random images (indexes of the selected images are provided in the Appendix Section) to perform the experiments. Using the ResNet50 FPN backbone, we extract P2 feature maps to create pairs of reference and distorted features. We utilize VVenc to distort the features with two different QP values (QP35 and QP41) to span various distortion ranges. We generated 1,000 distorted features from the 500 reference images.

C. TRAINING

The SR models are trained using a mini-batch size of 10. All the weights are randomly generated using a Gaussian distribution with a mean of 0 and a standard deviation of 0.02. We use the ADAM optimizer [21] for training, with a initial learning rate of 1×10^{-4}, and momentum parameters $\beta_1 = 0.5, \beta_2 = 0.99$. The proposed SR models are trained and inferred using a single NVIDIA Tesla V100 SXM2 32 GB using the PyTorch [51] framework. It took approximately 12 hours to train the model for 200 epochs.

D. ANALYSIS OF THE OBJECTIVE FUNCTION

We compare our proposed loss function in Eq. 4 to the MSE loss to verify the effectiveness of our proposed loss function. Table 1 shows the mAP inference results. Table 1 demonstrate that our proposed loss function consistently outperforms the L2 loss for all models. RCAN shows the best performance for most QP levels, followed by EDSR and CARN.

E. RD-PERFORMANCE COMPARED WITH ANCHOR

The experimental results of comparing anchor with our proposed method (i.e., best performing model RCAN) are shown in Table 1. The results in Table 2 show that spatially reducing the feature size improves the bpp rate significantly. Our proposed method achieved a 50.31% BD-rate reduction compared with the anchor baseline. Figure 6 shows the comparison of FSR models with the anchor baseline. All the models were able to significantly outperform the anchor baseline, especially for the lower bpps.

F. FEATURE ROBUSTNESS

Features extracted by DNN show high tolerance against distortions compared to raw images [47]. The robustness of the features against various distortions is shown in Figure 7. We generate five noises (Artificial White Gaussian Noise (AWGN), blur, resizing, quantization distortions, and coding artifacts) that occur naturally in the compression pipeline [43]. We extract the P2-P6 feature maps, using a ResNet [19] FPN-based pre-trained backbone with standard COCO2017 [16] image dataset. Pretrained Mask-RCNN is used to evaluate the quality of the distorted features and

TABLE 6. Images selected from the COCO dataset for training and testing.

Indexes for Training Images									
0004322	0006871	0006873	0006896	0006914	0007050	0007090	0007150	0007167	0007220
0007221	0007251	0007253	0007274	0007298	0007318	0007367	0007414	0007519	0007535
0007539	0007566	0007570	0007583	0007625	0007642	0007650	0007685	0007697	0007713
0007726	0007753	0007757	0007781	0007852	0007890	0007934	0007938	0007946	0011297
0012916	0014886	0014938	0014966	0014986	0014998	0015006	0015008	0015017	0015036
0015049	0015050	0015055	0015062	0015096	0015110	0015113	0015157	0015165	0015197
0015203	0015249	0015274	0015318	0015345	0015354	0015386	0015391	0015394	0015427
0015438	0015472	0015485	0015496	0015559	0015564	0015569	0015610	0015658	0015725
0015750	0015794	0015809	0015843	0015846	0015850	0015947	0016005	0016063	0016082
0016089	0016109	0016113	0016114	0016125	0016180	0016206	0016209	0016238	0016251
0016273	0016297	0016317	0016342	0016343	0016360	0016364	0016382	0016470	0016496
0016525	0016546	0016575	0016593	0016606	0016629	0016673	0016677	0016689	0016701
0017095	0022861	0025192	0025195	0025241	0025282	0025316	0025388	0025414	0025485
0025506	0025521	0025535	0025549	0025550	0025552	0025559	0025566	0025595	0025625
0025628	0025668	0025673	0025695	0025723	0025727	0025729	0025743	0025746	0025747
0025797	0025803	0025812	0025853	0025860	0025994	0025997	0026025	0026033	0026049
0026052	0026069	0026097	0026162	0026165	0026166	0026176	0026227	0026233	0026266
0026274	0026292	0026294	0026320	0026356	0026359	0026363	0026376	0026454	0026497
0026506	0026507	0026551	0026577	0026584	0026611	0026647	0026697	0026730	0026731
0026768	0026780	0026783	0026829	0026893	0026939	0026958	0026978	0026992	0026997
0027037	0027046	0027055	0027070	0027075	0027108	0027149	0027157	0027175	0027215
0027298	0027302	0027343	0027353	0027395	0027424	0027433	0027440	0027466	0027478
0027482	0027852	0031597	0036941	0037396	0038439	0039455	0039468	0039535	0039570
0039589	0039597	0039643	0039663	0039698	0039716	0039718	0039733	0039811	0039852
0039871	0039905	0039948	0039971	0039993	0040064	0040068	0040069	0040071	0040085
0040100	0040114	0040130	0040144	0040210	0040251	0040274	0040286	0040304	0040317
0040399	0040426	0040433	0040449	0040466	0040497	0040535	0040557	0040583	0040602
0040621	0040643	0040681	0040685	0040711	0040746	0040866	0040884	0040891	0040938
0040971	0040988	0040995	0041022	0041027	0041077	0041082	0041087	0041097	0041105
0041110	0041138	0041233	0041247	0041271	0041279	0041284	0041305	0041311	0041356
0041389	0041398	0041404	0041453	0041489	0041507	0041603	0044467	0047659	0048047
0048139	0052949	0053370	0060677	0065258	0068623	0069342	0070334	0083556	0083599
0084447	0092145	0092585	0093332	0096288	0096654	0097878	0098090	0102655	0105719
0106978	0107885	0109277	0113089	0120388	0126415	0128324	0136977	0139796	0141139
0142321	0142824	0148181	0148188	0148860	0149381	0149835	0150819	0154607	0156939
0157067	0158863	0165522	0167696	0169893	0170731	0170931	0172574	0172622	0173997
0175756	0183806	0185719	0186009	0186697	0192894	0193895	0202298	0203460	0207548
0212268	0215534	0216244	0222696	0223682	0226256	0229782	0231222	0236539	0237780
0246999	0247071	0250385	0254775	0257807	0264998	0278796	0280911	0285628	0286234
0287907	0291930	0299715	0300895	0301362	0301554	0304854	0305986	0309491	0310035
0312204	0314500	0316155	0316575	0321006	0321705	0321973	0332781	0333106	0334074
0340103	0341006	0347019	0349734	0350425	0352901	0354316	0354491	0357235	0357386
0357771	0358523	0361527	0368729	0373218	0374266	0377576	0380427	0381821	0384157
0385145	0391410	0396404	0397735	0399091	0403135	0404437	0409512	0415856	0417594
0427471	0434519	0435334	0443391	0446565	0450182	0454538	0455476	0457275	0458568
0459933	0468542	0468670	0477749	0479057	0480001	0480345	0483742	0485097	0485485
0487925	0489266	0492489	0494175	0498733	0500765	0502593	0503431	0504498	0514191
0514207	0515505	0517780	0518293	0519205	0520812	0525633	0530317	0530855	0530863
0532333	0533864	0536400	0537206	0541570	0542257	0544713	0550555	0552276	0553561
0555686	0558992	0559021	0564386	0564432	0564570	0564589	0565077	0570297	0573655
Indexes for Test Images									
001000	002153	008021	009769	009891	015335	017627	018150	018837	022589
022935	023230	024610	025560	025593	027620	155341	161397	165336	166287
166642	169996	172330	172648	176606	176701	179765	180101	186296	250758
259382	267191	287545	287649	289741	293245	308328	309452	335529	337987
338625	344029	350122	389933	393226	395343	395633	401862	402473	402992
404568	406997	408112	410650	414385	414795	415194	415536	416104	416758
427055	428562	430073	433204	447200	447313	448448	452321	453001	458755
462904	463522	464089	468965	469192	469246	471450	474078	474881	475678
475779	537802	542625	543043	543300	543528	547502	550691	553669	567740
570688	570834	571943	573391	574315	575372	575970	578093	579158	581100

images. Note that the baseline accuracy for the Mask-RCNN on instance segmentation is 40.3 mAP.

For Gaussian blur, $\sigma = [1, 1.5, 3]$. The results show that in the case of low variance (*sigma*=1, *sigma*=1.5), distortion impacts feature more than the image domain, and in the case of high variance (*sigma*=3), mAP for the image domain lowers more severely than the features. As the feature size is 4× smaller than the image size, we employ Gaussian kernel window sizes of 21 and 5, respectively.

For AWGN, $\sigma = [1, 3, 5]$. Features have higher robustness against AWGN than images. Images show a considerable loss in mAP accuracy compared to features, as shown in Figure 7.

We employ the downscale factor of 2, 4, and 8 for resizing. The results demonstrate that SR noise (resizing) impacts images more than features, as illustrated in Figure 7.

We vary the distortion rate by using different bit-depth levels = [8, 4, 2] for quantization. The result in the feature domain reveals no significant loss in mAP performance, but the image domain indicates a considerable drop in mAP with 2-bit quantization, as shown in Figure 7.

For Coding Noise, we use VVenC [49] and VVdeC [50]. To distort the images and features, we experiment on several different QP = [QP32, QP40, QP47] levels. Figure 7 shows a considerable decrease in mAP in the image domain compared to the features domain for coding noise. We may conclude from this extensive experiment that features are more resistant to various distortions than images and thus provide a strong basis for using features in the F-VCM pipeline.

G. INPUT SHAPE OF SR FEATURES

We trained the SR model CARN to compare the effectiveness of using single-channel versus multi-channel features as input. We evaluated the model on the COCO dataset using a pre-trained Mask R-CNN ResNet50 backbone. The results in Table 3 reveal that multi-channel features significantly outperform single-channel features in terms of mAP, with a BD-rate gain of 59.27% compared to the anchor and a 21.81% gain over single-channel features. Figure 8 illustrates the visual quality of the restored multi-channel and single-channel features, showing that the multi-channel restored features are more similar to the target HR features.

H. PERFORMANCE ON LOW QP LEVEL

In some applications, such as medical imaging, maintaining high quality is more important than achieving high compression rates. To examine how our proposed method performs under these conditions, we evaluated it at the low QP level of 27. We trained a model using QP27 and compared its performance to a model trained on features from higher QP levels (35-45). Our results in Table 4 indicate that the model trained on QP27 performs significantly better than the model trained on higher QP levels. The bpp was 6.18 in all cases. The baseline mAP score was 0.7927, while the mAP score for our proposed method with EDSR was 0.7878. Figure 9 illustrates the visual quality of the restored features, revealing that the features restored by the model trained on QP27 are visually closer to the ground truth. The model trained on QP27 achieved a PSNR of 34.29 and an SSIM of 0.93, while the model trained on high QP values had a PSNR of 31.04 and an SSIM of 0.8406. These results demonstrate the effectiveness of our proposed method in maintaining feature quality at low QP levels.

I. COMPUTATIONAL COMPLEXITY

The inclusion of an SR module in the F-VCM pipeline may increase computational complexity. To assess the impact of this addition, we compared the increase in the framework's overall inference time due to the SR module. The results indicated that using the SR module increased the computational parameters by an average of 38.97% and the inference time of the VCM pipeline by 7.08%. These results are shown in Table 5, which also presents the inference time, model running time, number of parameters, and total number of FLOPS for the SR models and compares them to the baseline anchor. The anchor method has a relatively short running time, as it does not require reading a large amount of feature information from memory and does not include pre or post-processing steps. These factors contribute to the efficient running time of the anchor method. These results provide insights into our proposed method's computational complexity and efficiency.

V. CONCLUSION

This paper proposes an F-VCM pipeline for preserving machine vision task performance by compressing features rather than images. We propose that features, as opposed to images, are more resistant to distortions, and thus transferring features through the compression pipeline can prevent the loss of vital information. As a result, we downscale the encoder features and upscale them on the decoder side, resulting in high compression rates with no performance degradation. Experimental results demonstrate that the proposed method outperforms the anchor pipeline, which employs the VVC encoder. Our proposed method generalizes effectively for a wide range of QP noise levels. Compared to traditional upsampling methods, our model provides a high visual quality reconstruction of the features.

APPENDIX

A. ANCHOR SETTINGS

As shown in Figure 2, the VCM Feature Compression anchor process consists of feature extraction, feature rearrange, and quantization before encoding. For decoding, the process is reversed with the steps of dequantization, feature rearrange, and evaluation network for the specified machine vision task. In the feature extraction process, P-Layer features from P2 to P6 are extracted using the ResNet-FPN backbone network. In feature rearrange, the 256-channel feature map of each layer is spatially tiled into a single channel. Quantization then converts the feature maps, which have floating point values, to integer values between 0 and 255 using the minimum and maximum values of the entire feature map as reference. The feature maps are then converted to the YUV 4:0:0 format using FFmpeg 4.2.0. The features are encoded and decoded using VTM 12.0, after which the reverse encoding process is carried out, including inverse quantization to restore the feature maps to floating point values using the same minimum and maximum values used in the initial quantization step. In the feature rearrange step, the spatially tiled channels are separated. The feature maps for the P2 to P6 layers are reconstituted with 256 channels each, concatenated in the channel direction. Finally, the evaluation network performs

the specified machine vision task using the restored feature map.

B. COCO DATASET SETTINGS

We selected 500 images from the COCO dataset for training the model with instance segmentation task. Table 6 lists the indexes of the COCO dataset images used for training and testing. These images were carefully chosen to ensure an even distribution of object classes.

ACKNOWLEDGMENT

(Jung-Heum Kang and Muhammad Salman Ali contributed equally to this work.) This research was a result of a study on the NIPA.

REFERENCES

[1] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset V4: Unified image classification, object detection, and visual relationship detection at scale," 2018, *arXiv:1811.00982*.

[2] Y. Kim, J. Park, Y. Jang, M. Ali, T.-H. Oh, and S.-H. Bae, "Distilling global and local logits with densely connected relations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6290–6300.

[3] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[8] T. Kumar, J. Park, M. S. Ali, A. F. M. S. Uddin, J. H. Ko, and S.-H. Bae, "Binary-classifiers-enabled filters for semi-supervised learning," *IEEE Access*, vol. 9, pp. 167663–167673, 2021.

[9] *[VCM] MPEG-VCM White Paper*, Standard m57491, Jul. 2021.

[10] H. Choi and I. V. Bajić, "Near-lossless deep feature compression for collaborative intelligence," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process. (MMSP)*, Aug. 2018, pp. 1–6.

[11] M. S. Ali, T. B. Iqbal, K.-H. Lee, A. Muqeet, S. Lee, L. Kim, and S.-H. Bae, "ERDNN: Error-resilient deep neural networks with a new error correction layer and piece-wise rectified linear unit," *IEEE Access*, vol. 8, pp. 158702–158711, 2020.

[12] Z. Chen, K. Fan, S. Wang, L. Duan, W. Lin, and A. C. Kot, "Toward intelligent sensing: Intermediate deep feature compression," *IEEE Trans. Image Process.*, vol. 29, pp. 2230–2243, 2020.

[13] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGPLAN Notices*, vol. 52, no. 4, pp. 615–629, May 2017.

[14] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[15] I. V. Bajic, W. Lin, and Y. Tian, "Collaborative intelligence: Challenges and opportunities," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 8493–8497.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[17] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[18] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2110–2118.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[20] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[22] W. Tan, B. Yan, and B. Bare, "Feature super-resolution: Make machine see more clearly," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3994–4002.

[23] C. Deng, M. Wang, L. Liu, Y. Liu, and Y. Jiang, "Extended feature pyramid network for small object detection," *IEEE Trans. Multimedia*, vol. 24, pp. 1968–1979, 2022.

[24] C. Dong, C. Loy, H. Change, and T. X. Kaiming, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[25] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.

[26] H. Choi and I. V. Bajic, "Deep feature compression for collaborative object detection," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3743–3747.

[27] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 769–776.

[28] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 341–349.

[29] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[30] K. Zhang, L. Van Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3217–3226.

[31] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 252–268.

[32] Z. Chen, L.-Y. Duan, S. Wang, W. Lin, and A. C. Kot, "Data representation in hybrid coding framework for feature maps compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3094–3098.

[33] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 517–532.

[34] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1664–1673.

[35] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 286–301.

[36] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[37] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 136–144.

[38] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.

[39] J. Johnson, A. Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[40] I. Goodfellow, J. Pouget-Abadie, M. Mirza, and X. Mehdi, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–10.

[41] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2

[42] L. Yu, *Draft Description of Exploration Experiments on Feature Compression for VCM*, Standard ISO/IEC JTC 1/SC 29/WG 2, m58290, Oct.2021.
[43] S. Liu, M. Rafie, and Y. Zhang, *Common Test Conditions and Evaluation Methodology for Video Coding for Machines*, Standard ISO/IEC JTC 1/SC 29/WG2, N00107, Jul. 2021.
[44] A. Wieckowski. *Update on Open Optimized VVC Implementations VVenC and VVdeC*, document NET-U0135, Joint Video Experts Team (JVET), 2021.
[45] H. Choi and I. V. Bajic, "Near-lossless deep feature compression for collaborative intelligence," 2018, *arXiv:1804.09963*.
[46] M. Lee and H. Choi, *[VCM Track 1] EE1.2: P-Layer Feature Map Anchor Generation for Object Detection on OpenImageV6 Dataset*, Standard ISO/IEC JTC 1/SC 29/WG 2, m58786, Jan. 2022.
[47] J. Kang, H. Heum, J. Won, K. Chang, M. Choi, A. Salman, B. Sung-Ho, and Y. K. Hui, "An analysis on the properties of features against various distortions in deep neural networks," *J. Broadcast Eng.*, vol. 26, no. 7, pp. 868–876, 2021.
[48] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[49] A. Wieckowski, J. Brandenburg, T. Hinz, C. Bartnik, V. George, G. Hege, C. Helmrich, A. Henkel, C. Lehmann, C. Stoffers, I. Zupancic, B. Bross, and D. Marpe, "VVenC: An open and optimized VVC encoder implementation," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2021, pp. 1–2.
[50] A. Wieckowski, G. Hege, C. Bartnik, C. Lehmann, C. Stoffers, B. Bross, and D. Marpe, "Towards a live software decoder implementation for the upcoming versatile video coding (VVC) codec," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3124–3128.
[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, and T. L. Killeen, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8024–8035.

JUNG-HEUM KANG received the bachelor's degree from the Department of Computer Science and Software Engineering, Ajou University, Suwon, South Korea, in 2020, and the M.S. degree from the Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea. His research interests include image processing, computer vision, video coding, machine learning, and international standardization.

MUHAMMAD SALMAN ALI received the bachelor's degree in computer science from the National University of Sciences and Technology (NUST), Islamabad, Pakistan. He is currently pursuing the M.S. degree leading to the Ph.D. degree with the Department of Computer Science and Engineering, Kyung Hee University, South Korea. His research interests include image compression, video coding, and machine learning.

HYE-WON JEONG received the bachelor's degree from the Department of Software Convergence, Kyung Hee University, Yongin, South Korea, in 2022, where she is currently pursuing the M.S. degree with the Department of Computer Science and Engineering. Her research interests include image processing, video coding, computer vision, machine learning, and international standardization for image and video coding.

CHANG-KYUN CHOI received the bachelor's and M.S. degrees from the Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea. His research interests include image processing, video coding, computer vision, machine learning, and international standardization for image and video coding.

YOUNHEE KIM received the B.S. and M.S. degrees in computer science from Ajou University, Suwon, Republic of Korea, in 2000 and 2002, respectively, and the Ph.D. degree in computer science from George Mason University, Fairfax, VA, USA, in 2009. She has been a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea, since 2009. Her current research interests include video coding, image and video signal processing, and computer vision.

SE YOON JEONG received the B.S. and M.S. degrees from Inha University, in 1995 and 1997, respectively, and the Ph.D. degree from the Korea Advanced Institute of Science and Technology, in 2014. In 1996, he joined the Electronics and Telecommunications Research Institute, as a Principal Researcher. He has many contributions to the development of international standards, such as scalable video coding and highly efficient video coding. His current research interests include video coding for machine- and AI-based video coding.

SUNG-HO BAE (Member, IEEE) received the B.S. degree from Kyung Hee University, South Korea, in 2011, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012 and 2016, respectively. From 2016 to 2017, he was a Postdoctoral Associate with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. Since 2017, he has been an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University. His research interests include model compression/interpretation for deep neural networks and inverse problems in image processing and computer vision.

HUI YONG KIM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Department of Electrical and Electronic Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1994, 1998, and 2004, respectively. He was a Multimedia Team Leader with the Adpark Technology Research Institute, in 2005. From 2005 to 2019, he was the Group Director/Responsible Researcher with the Realistic AV Research Group, Korea Electronics and Telecommunications Research Institute (ETRI). From 2019 to 2020, he was an Associate Professor with the Department of ICT Convergence Engineering, Department of Electronic Engineering, Sookmyung Women's University, Seoul, South Korea. He is currently an Associate Professor with the Department of Computer Science and Engineering, Kyung Hee University, Yongin, South Korea. His current research interests include video coding/international standardization and inverse problems in image processing and computer vision using deep neural networks.