IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 31 January 2023, accepted 5 March 2023, date of publication 9 March 2023, date of current version 20 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3254589

RESEARCH ARTICLE

Feature Map Compression for Video Coding for Machines Based on Receptive Block Based Principal Component Analysis

MINHUN LEE[1], HANSOL CHOI[1], JIHOON KIM[1], JIHOON DO[2], HYOUNGJIN KWON[2], SE YOON JEONG[2], DONGGYU SIM[1], (Senior Member, IEEE), AND SEOUNG-JUN OH[3], (Member, IEEE)

[1]Department of Computer Engineering, Kwangwoon University, Seoul 139701, South Korea
[2]Media Coding Research Section, Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea
[3]Department of Electronic Engineering, Kwangwoon University, Seoul 139701, South Korea

Corresponding authors: Donggyu Sim (dgsim@kw.ac.kr) and Seoung-Jun Oh (sjoh@kw.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under a grant funded by the Korean Government through Ministry of Science, ICT (MSIT) (Video Coding for Machine) under Grant 2020-0-00011, in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) under a grant funded by the MSIT and Future Planning under Grant NRF-2021R1A2C2092848 and in part by the Excellent Researcher Support Project of Kwangwoon University in 2021.

ABSTRACT This paper presents a method to effectively compress the intermediate layer feature map of a convolutional neural network for the potential structures of Video Coding for Machines, which is an emerging technology for future machine consumption applications. Notably, most extant studies compress a single feature map and hence cannot entirely consider both global and local information within the feature map. This limits performance maintenance during machine consumption tasks that analyze objects with various sizes in images/videos. To address this problem, a multiscale feature map compression method is proposed that consists of two major processes: receptive block based principal component analysis (RPCA) and uniform integer quantization. The RPCA derives the complete basis kernels of a feature map by selecting a set of major basis kernels that can represent a sufficient percentage of global or local information according to the variable-size receptive blocks of each feature map. After transforming each feature map using the set of major basis kernels, a uniform integer quantizer converts the 32-bit floating-point values of the set of major basis kernels, corresponding RPCA coefficients, and a mean vector to five-bit integer representation values. Experiment results reveal that the proposed method reduces the amount of feature maps by 99.30% with a loss of 8.30% in the average precision (AP) on the OpenImageV6 dataset and 0.77% in AP_M and 0.47% in AP_L on the MS COCO 2017 validation set while outperforming previous PCA-based feature map compression methods even at higher compression rates.

INDEX TERMS Moving picture experts group, video coding for machines, convolutional neural network, principal component analysis, feature map compression.

I. INTRODUCTION

Over the past few decades, image/video data generated by sensors have become the most used data sources worldwide. In addition, owing to the rapid growth in machine-learning applications based on video data, the volume of image/video data has rapidly increased. Accordingly, the current volume of video data used by machines exceeds that used by humans [1]. This is because machine consumption tasks, such as object detection, segmentation, tracking, and other machine-based applications, use data differently from humans. In addition, several deep learning (DL)-based studies are being actively conducted for various machine consumption tasks that are applicable to smart applications

The associate editor coordinating the review of this manuscript and approving it for publication was Yue Zhang.

and services, including autonomous vehicles, the Internet of things, intelligent closed-circuit television systems, and smart cities [2], [3], [4], [5], [6]. Typically, computers require massive volumes of video data to perform intelligent analysis. However, such data need to be compressed before transmission to reduce the transmission time and cost. Because humans use video data for various purposes, such as streaming and entertainment, developing video data compression methods using human visual characteristics with high objective and subjective qualities are considered a top priority. In particular, the traditional Moving Picture Experts Group (MPEG) video coding standard techniques, such as Advanced Video Coding (AVC) [7], High Efficiency Video Coding (HEVC) [8], and Versatile Video Coding (VVC) [9], are common video data compression methods that significantly improve the video coding efficiency by squeezing out the spatiotemporal pixel-level redundancy in video frames.

However, in the field of image/video analytics, the ability of existing video coding standard tools in performing visual signal level compression to efficiently process such data is questionable, as data analysis is performed by computers rather than by humans. Therefore, a new compression scheme that is capable of preserving information, rather than preserving image quality for human consumption, is required so that machines can accurately recognize image information. Thus, MPEG launched a formal activity on a new standardization called Video Coding for Machines (VCM) during their 127th meeting. Furthermore, to assess the availability of adequate evidence for the standardization task, the group issued its call for evidence document [10]. Notably, MPEG-VCM aims to define an efficient bitstream format generated by compressing videos or its corresponding feature maps in DL networks after decompression to perform multiple tasks while preserving machine consumption task performance across various applications. Its ultimate purpose is to compress a feature map or image/video for machine consumption or hybrid machine–human consumption. To this end, three potential structures are suggested in the standardization process of MPEG-VCM, and these are illustrated in Fig. 1. The structures shown in Figs. 1 (a) and (b) are considered potential structures for machine consumption tasks. The potential structure illustrated in Fig. 1 (a) compresses the feature map extracted from an arbitrary layer in a DL network while the remaining network obtains the result of the machine consumption task. This paper presents a discussion on a layer from which the feature map is extracted. The second potential structure is shown in Fig. 1 (b) compresses an image/video, and a DL network uses the reconstructed data. Both the foregoing structures can reduce the volume of data in feature maps and avoid performance degradation during machine consumption tasks. A potential structure for hybrid machine-human consumption tasks is displayed in Fig. 1 (c). This structure consists of a combination of structure for the machine consumption task as shown in Figs. 1 (a) and (b) and a structure for the human consumption task. In Fig. 1 (c), the video and feature map encoders are designed to optimize both the compression ratio and performance for human and machine consumption tasks, respectively. Moreover, when compressing and transmitting feature map data, as shown in Figs. 1 (a) and (c), only the machine consumption task segment of the DL network on the decoded feature maps can offload part of the computation from the server to the front-end device, benefiting in terms of the computational latency and energy consumption [11]. Therefore, the compression of feature map data is important for applications on edge devices, such as mobile and embedded devices. Currently, the standardization activities of MPEG-VCM involve collecting evidence focusing on machine-only consumption tasks.

Among the various machine consumption tasks, the following five tasks that are common to various VCM service scenarios: object detection, instance segmentation, object tracking, action recognition, and pose estimation are determined as the main tasks. Accordingly, the following five DL networks: Faster R-CNN X-101 feature pyramid network (FPN) [12], [13], [14], Mask R-CNN X101-FPN [13], [14], [15], JDE-1088 $\times$ 608 [16], Slowfast [17], and HRNet [18] are determined. Especially, object detection is an important machine consumption task that deals with the detection instances of visual objects belonging to a certain class. Also, it is an actively researched area, for which various popular datasets and benchmarks have been released. Therefore, this work focuses on object detection tasks. The Faster R-CNN X101-FPN uses a pyramid scheme that generates multiple feature maps from input data. This structure, called the FPN structure, utilizes multilayer feature maps and is used in various DL networks, including most of the primary tasks of MPEG-VCM, such as the Faster R-CNN X101-FPN, Mask R-CNN X101-FPN, and JDE-1088 $\times$ 608. Generally, the FPN structure is used as a feature extractor in a DL network [14]. Thus, X101-FPN is defined as a feature extractor of Faster R-CNN X101-FPN. In addition, the FPN feature maps generated by X101-FPN are define as targets for compression, and they are consisted of four different scale feature maps called $\{P_2, P_3, P_4, P_5\}$ as shown in Fig. 2. Moreover, there exist 256 channels in each P_i. The comparison of the total output feature map data of X101-FPN and input image determined by MPEG-VCM for evaluating the object detection task revealed that the total feature map data were significantly greater than the input image.

Several methods have been proposed to compress feature map data. However, most of them compressed single feature maps rather than multiscale feature maps similar to FPN feature maps. Moreover, research on compressing feature maps extracted from the FPN structure in MPEG-VCM is still in the nascent stages. Even when compressing multilayer feature maps with the latest video coding standard, VVC, a significant loss of precision is observed in machine consumption tasks at high compression rates. One reason for this is that compression is performed based on

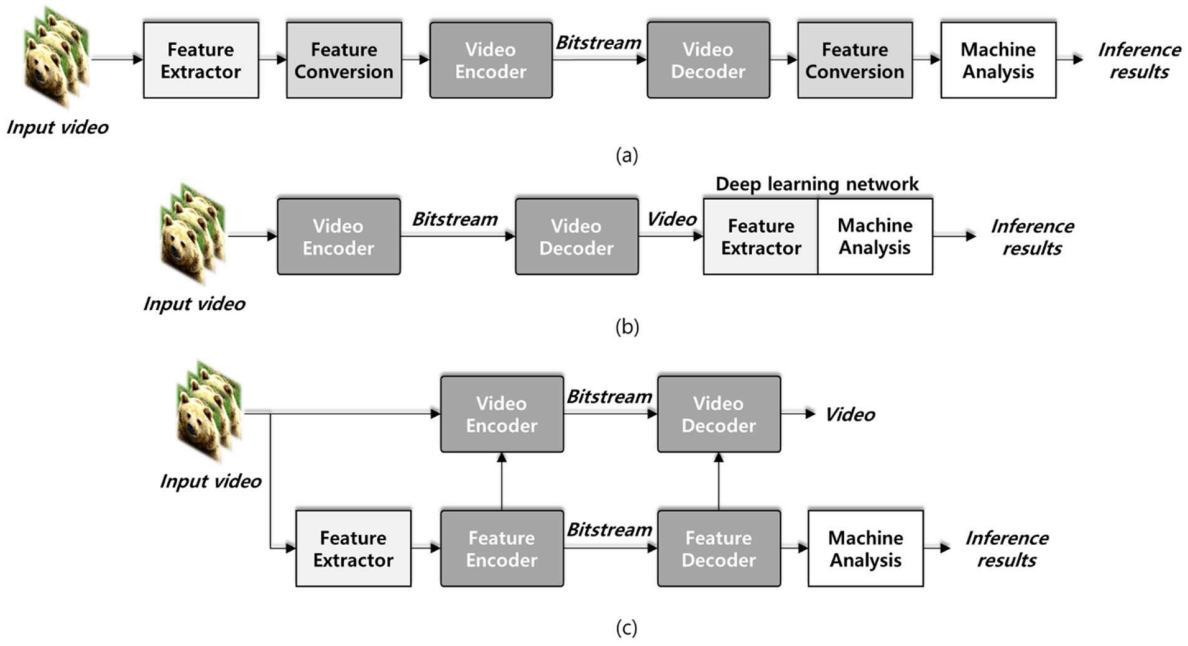

FIGURE 1. Potential structures for MPEG-VCM. (a) First potential structure for machine consumption task. (b) Second potential structure for machine consumption task. (c) Third potential structure for hybrid machine-human consumption tasks.

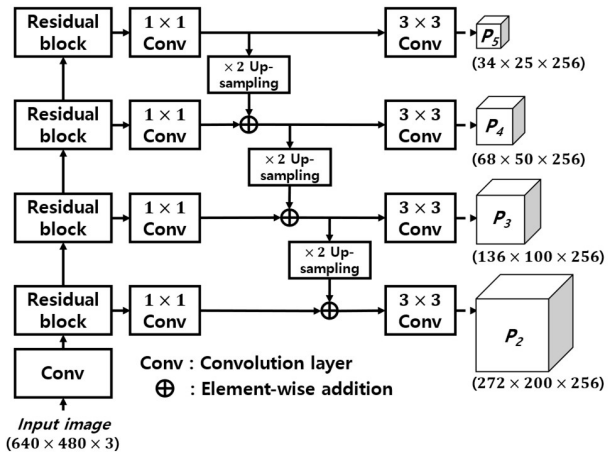

FIGURE 2. X101-FPN structure.

rate-distortion optimization (RDO), where human consumption is considered rather than machine consumption. Another reason is that the feature map data are considerably greater than the input image data because the multiple channels of feature maps are generated based on the three-channel input image. Each channel in the feature map, which is the output of the *k-th* convolution layer data, is extracted from the input of the *k-th* convolution layer using each filter of the convolution layer, therefore, each channel of the feature map is treated as possibly independent [19]. Therefore, FPN feature maps with 256 channels represent an input image expressed in the form of three-channel with 256 characteristics through convolution filters in the convolution layer. However, it may lead to significant redundancy in each channel and varying level of importance for the required machine consumption task [20]. Consequently, reducing this redundancy is necessary to efficiently compress feature map data. Based on these properties, Son and Kim [21] proposed a single layer feature map compression method for an object detection task in an MPEG meeting. In particular, the compression target feature map was extracted from the intermediate layer of YOLO9000 [22], and principal component analysis (PCA) [23] was performed to reduce the dimensionality of the feature map. In addition, several methods for feature map compression based on PCA have been proposed [24], [25]. These methods were also compressed a single layer feature map extracted from a shallow DL network. However, each FPN feature map extracted from the pyramidal neural network has a different channel size for use in detecting objects of different sizes. More specifically, P_2 with large channel size are used for small object detection, and P_5 with small channel size used for large object detection. Let's call the block of the same size as the channel in P_5 a receptive block.

Accordingly, feature maps with small channel sizes have a large receptive block, and vice versa. Consequently, the FPN feature maps can be used the simultaneous detection of multiple objects on a wide scale. Therefore, it is important to preserve local and global information by considering the receptive block of each feature map during dimensionality reduction of FPN feature maps. However, performing PCA on the entire FPN feature maps from X101-FPN may be inappropriate as previous methods [21], [24], [25] do not consider the receptive block of the feature map and compress a single layer feature map from shallow DL networks, such as YOLO9000, ResNet-18, ResNet-34 [26], and MobileNetV2 [27].

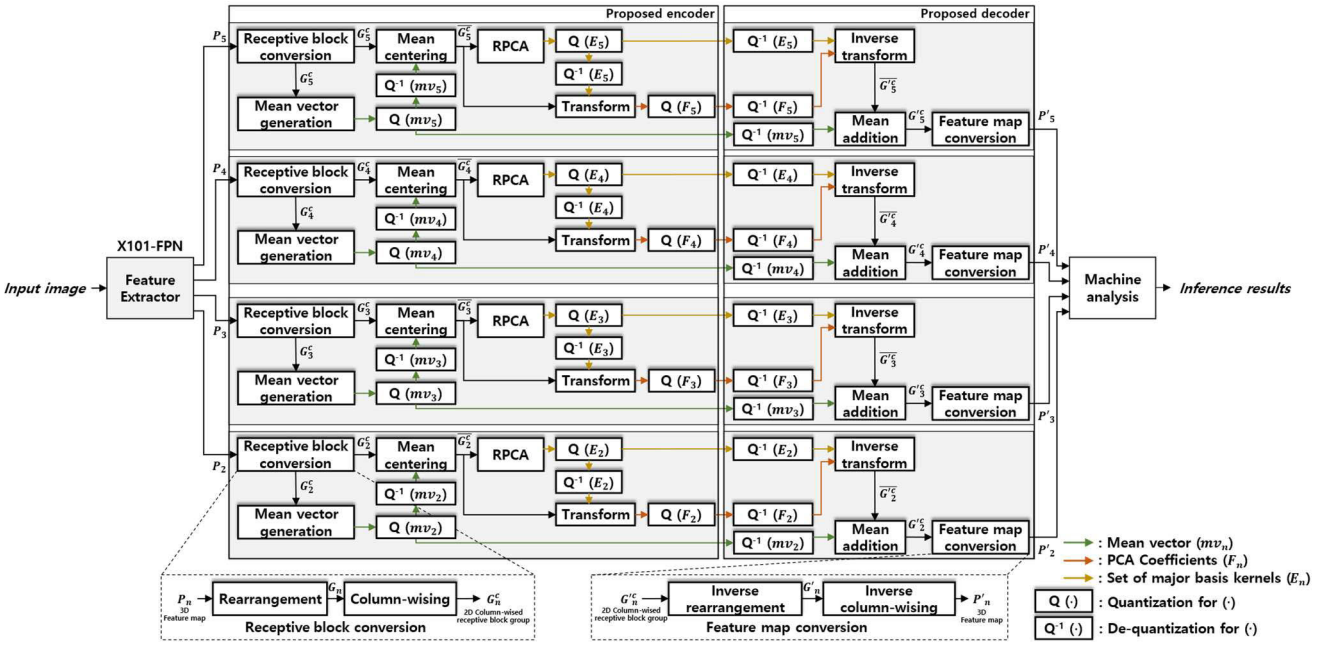

FIGURE 3. Proposed feature map compression method consisting of receptive block based PCA (RPCA) and quantization processes.

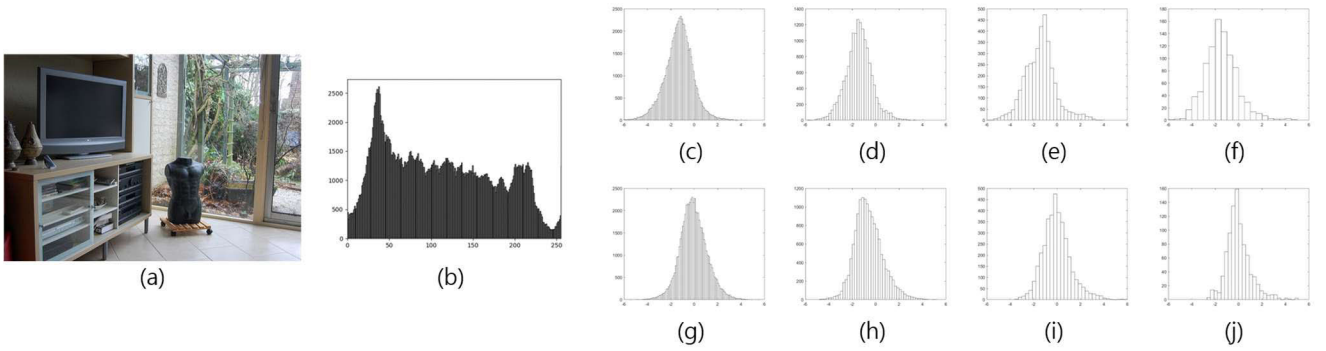

FIGURE 4. Histogram of an original image in the MS COCO 2017 dataset and feature maps. (a) Original image. (b) Grayscale histogram of the original image. (c), (d), (e), and (f) Histogram of the 80th channel of P_2 to P_5. (g), (h), (i), and (j) Histogram of the 120th channel of P_2 to P_5.

With this background, this paper proposes an effective PCA-based compression method for feature maps that are outputs of the feature extractor modules displayed in Fig. 1 (a) and (c). The proposed block-based PCA method reconstructs the feature map while preserving both the global and local information of various size objects within each FPN feature map. Therefore, the proposed method performs efficient compression while maintaining the precision of machine consumption tasks in MPEG-VCM. More specifically, it achieves a higher precision with a smaller volume of data, which is only 0.7% of the volume of the original FPN feature maps compared to previous methods. The proposed method can also be used as a pre/post-processing method to be used with other feature map compression methods based on potential structures under consideration in MPEG-VCM, as displayed in Figs. 1 (a) and (c).

Note that this work is an extension of our previous contribution to the 132nd MPEG meeting [28]. The remainder of this paper is organized as follows. Section II briefly reviews related research. Section III details the proposed method. The experimental results are presented and discussed in Section IV to validate the effectiveness of the proposed scheme. Finally, Section V concludes the paper.

II. RELATED RESEARCH

Notably, each value of a feature map extracted from the intermediate layer of a DL network with input image/video represented by an eight-bit integer value is expressed as a 32-bit floating-point real number. The feature map has significantly more channels than the input data. Therefore, it contains more volume of data than the input data; in particular, for multilayer feature maps, the number of feature maps is much greater than the input data. Hence, the compression of feature maps for machine consumption rather than image/video is much more difficult.

Several methods for data compression have been proposed for machine consumption tasks by compressing a single layer feature map. In particular, Choi and Bajić [29] and Yoon and

Kim [30] proposed a lossy feature map compression method. The former method compressed the feature map from the 17th max-pooling layer of YOLO9000 through quantization and HM-16.12 (HEVC test model). Consequently, they discovered that lossy compression through HM-16.12 and not the eight-bit uniform quantized data significantly impacted object detection task precision. Therefore, they proposed compression-augmented training to prevent precision degradation with increasing quantization parameters. Furthermore, Yoon and Kim [30] focused on the effect of feature map quantization with under eight-bit on the object detection task precision and performed channel-wise normalization of feature maps before quantization using four bits. In addition, Fischer et al. [31] proposed compressing an image through VTM-8.0 (VVC test model) and performed an object detection task with the reconstructed image. Because the existing video codec, including VTM-8.0, has been designed for human consumption tasks, the authors proposed a feature-based RDO method for machine consumption tasks. The encoder achieved high feature fidelity using the distortion in the feature domain output from the first layer of the DL network instead of the distortion in the pixel domain in the RDO process. Furthermore, Xia et al. [32] and Yang et al. [33] proposed a compression scheme for both machine and human consumption tasks to reduce the volume of data transmitted for the latter. Particularly, Xia et al. [32] extracted the keypoint features from every frame and compressed them for the action recognition task. Here, only the keyframe was compressed and transmitted to reduce transmission data for the human consumption task. The middle frame was generated through a generative model with the guidance of keypoint features and keyframes. Yang et al. [33] compressed and transmitted only the compact structure and color features extracted from the input image and performed face landmark detection using the reconstructed feature map data. Images for the human consumption task were generated on the decoder side. In addition, Kim et al. [34] and Shao et al. [35] proposed transform-based feature map compression methods using an 8×8 discrete cosine transform (DCT) in common. Particularly, Kim et al. [34] transformed a fully connected layer in VGG16 [36] using an 8×8 DCT and subsequently quantized the DCT coefficients using uniform quantization. Shao et al. [35] transformed the single layer feature map obtained from shallow DL networks to significantly reduce the required on-chip memory size and off-chip memory access bandwidth. More importantly, DCT coefficients for the feature map were quantized using two-steps quantization based on a JPEG Q-table and encoded by storing only the nonzero coefficients.

As can be inferred, most previous studies on feature map compression methods focused on the feature map data extracted from a single layer in shallow DL networks. However, to apply feature map compression to various applications for multiscale object detection, efficient compression methods for multilayer feature maps extracted from networks such as FPN structure should be considered. As mentioned in Section I, the FPN structure has been used in various DL networks in MPEG-VCM. Therefore, this paper proposes a method for compressing the FPN feature maps extracted from X101-FPN based on Fig. 1 (a) for object detection tasks. Herein, the feature maps are compress by performing a block-based PCA and quantization processes on each FPN feature map.

III. PROPOSED FEATURE MAP COMPRESSION METHOD

This section presents the proposed method for efficient feature map compression. First, the part of the proposed scheme is briefly introduced. Each part is detailed in the remaining subsections. As shown in Fig. 3, the proposed feature map compression structure performs sequential block-based PCA and uniform integer quantization. Typically, in the image/video compression process, DCT with excellent energy compaction properties is used as a key technique [37]. However, as highlighted in Fig. 4, FPN feature maps and input image possess different characteristics. Unlike image, most histograms of the FPN feature maps exhibit a Gaussian distribution. The FPN feature maps of all channels reveal similar distributions for all feature maps with a Gaussian distribution as shown in Fig. 4. For Gaussian sources, the Karhunen-Loève transform, an orthonormal transform that produces uncorrelated transform coefficients, yields an optimal transform [38]. Therefore, PCA is employed to extract the basis kernels for efficient reduction in the dimensionality of the FPN feature maps to be transmitted by eliminating the redundancy between correlated data in each feature map.

Conventional PCA-based feature map compression methods compressed a single layer feature map [24], [25]. Those methods cannot efficiently be applied to the FPN feature map since it has a different channel size for use in detecting objects of different sizes. Therefore, a feature map compression method is proposed that applies block-based PCA to a zero-mean column-wised receptive block group. This process is called receptive block based PCA (RPCA). The zero-mean column-wised receptive block group for each FPN feature map is obtained by performing mean-centering on a column-wised receptive block group which is generated by the receptive block conversion process as shown in Fig. 3. This process consists of two steps as shown in Fig. 5. Each FPN feature map is firstly divided into the receptive block to generate a receptive block group. After that, the receptive block group is reshaped into a column-wised receptive block group which is explained later in detail. In addition, since the data to be transmitted for reconstruction of the FPN feature maps is expressed by 32-bit floating-point real values, a uniform integer quantization process is performed for further compression.

As mentioned before, the size of the receptive block is the channel size of P_5. Therefore, a set of basis kernels in unit of the receptive block is derived by the RPCA. Subsequently, a set of major basis kernels containing an adequate percentage of both global and local information in

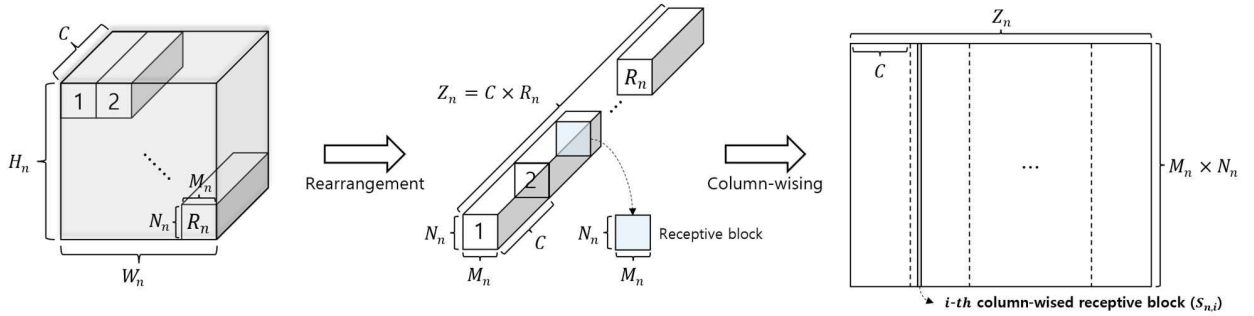

FIGURE 5. Receptive block conversion process consisting of rearrangement and column-wising from each FPN feature map to column-wised receptive block group $\mathbf{G_n^c}$ **(n = 2, 3, 4, 5).**

variable-size receptive blocks of each FPN feature map is selected from all the basis kernels for each FPN feature map based on the explained variance ratio. Following this, the set of major basis kernels is used to transform each FPN feature map. More specifically, the de-quantized set of major basis kernels is used in transform to prevent mismatch because they will be transmitted. Finally, the set of major basis kernels, corresponding RPCA coefficients, and the mean vector of each FPN feature map are obtained. The obtained RPCA coefficients and the mean vector are also independently quantized using the uniform integer quantizer for each data to additional compression. The proposed method can perform effective feature map compression on the output of the feature extractor modules of the potential structures of MPEG-VCM.

A. RECEPTIVE BLOCK BASED PRINCIPAL COMPONENT ANALYSIS

The primary purpose of the object detection task is to accurately detect objects with varying sizes, from large to small, within an image. The FPN structure uses a pyramid hierarchy structure to construct high-level semantic feature maps on all scales. maps on all scales. Therefore, the 2D block-based PCA is applied to each column-wised receptive block group after each P_n is rearranged into the corresponding group shown in Fig. 5. The optimal basis kernels for each group are independently derived efficiently to represent its local as well as global information.

Each P_n is rearranged by dividing it into receptive blocks of $M_n \times N_n$ and then concatenating them to construct a 3D receptive block group G_n, where $n \in \{2, 3, 4, 5\}$, $G_n \in \mathbb{R}^{M_n \times N_n \times Z_n}$, and $Z_n = C \times R_n$. After that, each 3D receptive block group is column-wised to obtain a 2D column-wised receptive block group G_n^c, where $G_n^c \in \mathbb{R}^{(M_n \times N_n) \times Z_n}$. Let $S_{n,i} \in \mathbb{R}^{(M_n \times N_n) \times 1}$ denote the *i-th* column-wised receptive block of G_n^c, where $i \in \{1, \ldots, Z_n\}$. After constructing the 2D column-wised receptive block groups, a mean vector is subtracted from each column-wised receptive block $S_{n,i} \in G_n^c$. The mean vector, $mv_n \in \mathbb{R}^{(M_n \times N_n) \times 1}$ of G_n^c, is computed as following:

$$mv_n = \frac{1}{Z_n} \sum_{i=1}^{Z_n} S_{n,i}. \tag{1}$$

Then, the zero-mean G_n^c denoted by $\overline{G}_n^c$ can be obtained by $S_{n,i} - mv_n$. PCA is independently applied to each $\overline{G}_n^c$ to obtain basis kernels and its associated set of eigenvalues: E_n and Λ_n.

$$\Lambda_n, E_n = PCA\left(\overline{G}_n^c\right). \tag{2}$$

At the first step in of $PCA(\cdot)$, the eigenvalues for $\overline{G}_n^c$ and their associated basis kernels are computed and stored in $\bar{\Lambda}_n$ and $\bar{E}_n$, respectively. After they are sorted in descending order, the eigenvalues in $\bar{\Lambda}_n$ and the associated basis kernels in $\bar{E}_n$ are stored in Λ_n and E_n, respectively. Then, first Q_n eigenvalues in Λ_n are chosen as principal components where $B_n \geq Q_n$ and B_n is the size of Λ_n. The number of principal components, Q_n, is typically determined by a cumulative explained variance ratio (CEVR) [39]. V_{Q_n}, the CEVR of G_n^c, is defined as following:

$$V_{Q_n} = \frac{\sum_{x=1}^{Q_n} \lambda_n^x}{\sum_{y=1}^{B_n} \lambda_n^y}, \tag{3}$$

where λ_n^x denotes eigenvalues in Λ_n. Thus, the value of V_{Q_n} is equal to the proportion of eigenvalue attributed to each selected major basis kernel. In this paper, V_{Q_n} is empirically determined and is detailed in the next section.

Thereafter, the RPCA coefficients F_n of each P_n is obtained by performing transform with $\overline{G}_n^c$ using the de-quantized set of basis kernels selected according to V_{Q_n}. Then, the RPCA coefficients and mean vectors are also quantized, respectively. This quantization process is detailed in the next section. Thus, $\overline{G'}_n^c$ is reconstructed using a de-quantized E_n and a de-quantized F_n, then, reconstructed G'^c_n is obtained by adding de-quantized mv_n to each reconstructed zero-mean column-wised receptive block. Then, feature map conversion which is backward process of receptive block conversion is performed to generate reconstructed P'_n.

P_5 is the smallest feature map in the FPN structure which may preserve the global information for detecting large objects. Thus, the size of P_5 is chosen as the receptive block. Other larger size feature maps such as P_2, P_3, and P_4 are divided into the receptive block size since it can be assumed that local information is preserved to detect both medium and small objects. Therefore, RPCA can derive the optimal basis

kernels while preserving information within each FPN feature map. In addition, the selected major basis kernels based on the CEVR can efficiently reduce the dimensionality. Therefore, the reconstructed feature maps share similar characteristics with the original feature maps, which are conducive to the detection of objects with varying sizes.

B. UNIFORM INTEGER QUANTIZATION

The RPCA process is followed by a uniform integer quantizer. Since one of the goals of MPEG-VCM is to compress feature maps while maintaining the performance of machine consumption tasks with a small amount of data and integer representation should be used in MPEG standards up to now, it may be better that all data are expressed in the integer type instead of the floating-point type. Therefore, the 32-bit floating-point output data of RPCA and mean vector for each FPN feature map are quantized into integer data. Note that in this process, the degree of data compression by quantization depends on the value of q, which is the quantization bit-depth. However, performing quantization with a low bit-depth can lead to incurs severe accuracy degradation because of producing a great deal of quantization error. Therefore, the experiments are conducted under various conditions to choose q that can improve the compression rate while maintaining performance, which is discussed in more detail in the next section.

Algorithm 1 describes the quantization process of the proposed method. In the first step, the minimum and maximum values of the set of major basis kernels, corresponding RPCA coefficients, and the mean vector are computed per each feature map, respectively. This process is followed by min–max normalization to represent each value as a floating-point real number between 0 and 1. Then, each normalized value is multiplied by $(2^q–1)$ and round off to represent a q-bit integer value. The round off process accounts for the quantization error. Algorithm 2 presents the de-quantization process. This involves reconstructing an approximation of the original values of the transformed data by multiplying the quantized value with the same quantization step, qs, as shown below:

$$qs = (max - min)/(2^q - 1), \tag{4}$$

where min and max denote the minimum and maximum values, respectively. min is added to express each value as a floating-point real number in the original range of each bit of the set of major basis kernels, corresponding RPCA coefficients, and the mean vector, respectively.

Thus, the FPN feature maps extracted from X101-FPN can be greatly compressed using RPCA and uniform integer quantization. In the next section, the objective quality will be evaluated of the proposed method.

Algorithm 1 Quantization

Input
$D(i)$: The set of major basis kernels, corresponding RPCA coefficients, and the mean vector
q: Quantization bit-depth
Output
$D'(i)$: Quantized $D(i)$
Begin
for $i = 0$ **to**size of $D(i)$
if $D(i) < min$ **then**
$min = D(i)$
end
if $D(i) > max$ **then**
$max = D(i)$
end
end
for $i = 0$ **to** size of $D(i)$
$D'(i) = (D(i) - min) / (max - min)$
$D'(i) = D'(i) \times (2^q - 1)$
$D'(i) = \text{round}(D'(i))$
end
End.

Algorithm 2 De-quantization

Input
$D'(i)$: Quantized $D(i)$
q: Quantization bit-depth
min: Minimum value of all $D(i)$
max: Maximum value of all $D(i)$
Output:
$\tilde{D}$(i): De-quantized $D'(i)$
Begin
for $i = 0$ **to** size of $D'(i)$
$\tilde{D}(\text{i}) = D'(i) \times \{(max - min) / (2^q - 1)\}$
$\tilde{D}(\text{i}) = D'(i) + min$
end
End.

IV. EXPERIMENTAL RESULTS

In this section, the performance of the proposed method is evaluated. The experimental setup is firstly introduced and then confirm the objective quality. In our experiment, the proposed method is implemented using Faster R-CNN X101-FPN, pre-trained on the MS COCO 2017 training set in Facebook AI Research Detectron2 [40]. In addition, 640 × 480 images from the MS COCO 2017 validation set [41] and 5K images from the OpenImageV6 dataset [42] which has been used for evaluation in MPEG-VCM, are used as the test datasets.

The volume of data is recorded to evaluate the objective quality. The object detection performance is quantified using the average precision (AP) value on both datasets. In addition, the performances of small, medium, and large objects are quantified in terms of their AP (AP_S, AP_M, and AP_L, respectively) to evaluate the detailed AP across varying scales in the MS COCO 2017 validation set. Furthermore, the defined values, R_{FM}, under each condition is calculated and compared to analyze the amount of compressed data by executing the transform and uniform integer quantization processes of the proposed method. Note that the value of R_{FM} also represents the ratio of the amount of compressed data to the amount of FPN feature maps extracted from X101-FPN

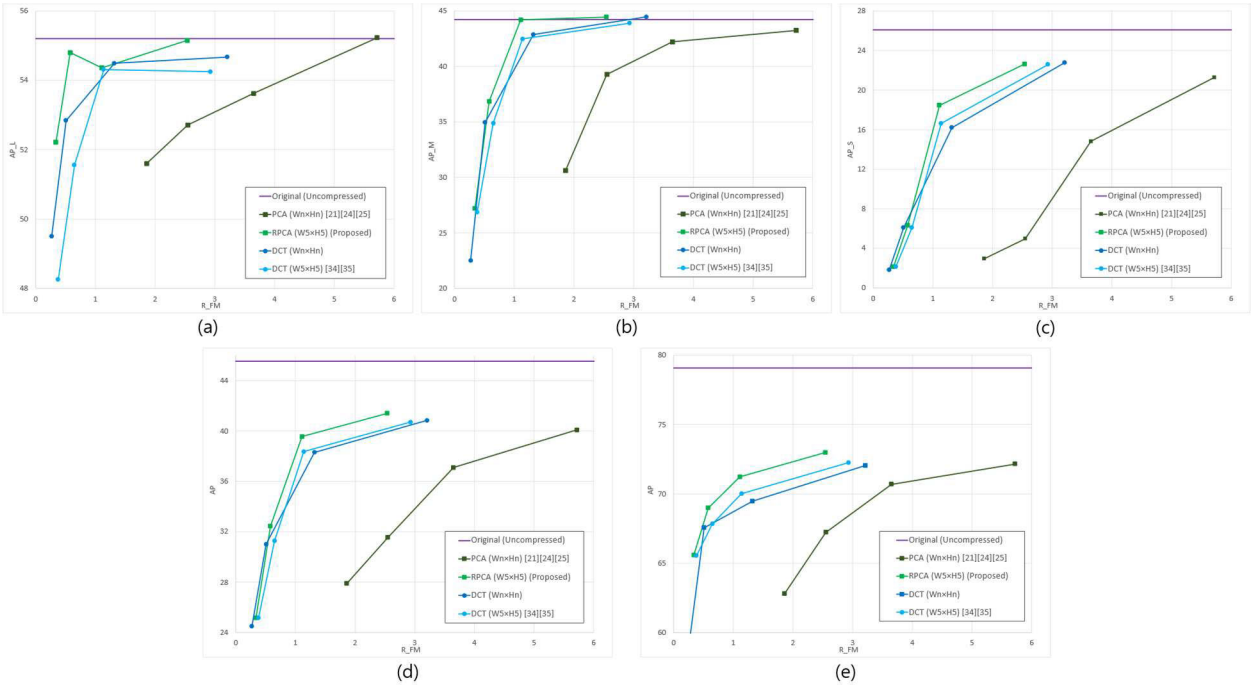

FIGURE 6. Average precision (AP) performance according to $\mathbf{R_{FM}}$ on each test dataset, (a)-(d) on MS COCO 2017 validation set, (e) on OpenImageV6 dataset. (a) $\boldsymbol{AP_L}$. (b) $\boldsymbol{AP_M}$. (c) $\boldsymbol{AP_S}$. (d) AP on MS COCO 2017 validation set. (e) AP on OpenImageV6 dataset.

before compression, which is defined as follows:

$$R_{FM} = \frac{D_{TEST}}{D_{FM}} \times 100\,(\%)\,, \tag{5}$$

where D_{TEST} represents the total volume of data compressed during the transform and uniform integer quantization that is used for reconstructing each FPN feature map, consisting of four sets of major basis kernels, corresponding RPCA coefficients, and four mean vectors for the proposed method. In addition, D_{FM} represents the total amount of data for the FPN feature maps. Therefore, a lower value of R_{FM} denotes a higher compression ratio.

Table 1 summarizes the performance of the proposed method according to the receptive block size and V_{Q_n}, which determines the number of major basis kernels to be selected when the quantization bit-depth is eight. In the case of the DCT-based methods, the number of DCT coefficients for each FPN feature map is determined according to V_{Q_n} in the same concept as the PCA-based methods. To compare the performance of the proposed method with that of the previous feature map compression methods, the transform and quantization processes are performed under each condition. More importantly, several experiments are conducted for a receptive block size of $W_n \times H_n$ for each FPN feature map, which is the same condition as that in the transform process involved in previous PCA-based feature map compression methods [21], [24], [25].

For all cases, the proposed uniform integer quantization is applied. Furthermore, to compare the previous DCT-based feature map compression methods [34], [35], the performance of the two transform methods for PCA-based transform and DCT is compared when using the same receptive block size for comparison. These will be referred to as "PCA-case" and "DCT-case," respectively. Although previous DCT-based methods [34], [35] perform 8×8 block-based DCT, the FPN feature maps may not have width or height that are multiples of eight depending on the input image size. Therefore, block-based DCT with $W_5 \times H_5$, which are the channel size of P_5, was performed for comparison. Meanwhile, D_{TEST} in (5) denotes the proportion of the four DCT coefficients for the DCT-case. As indicated in Table 1, the proposed method and the extant PCA-based approaches [21], [24], [25] present similar precision for all AP metrics, such as AP_S, AP_M, AP_L, and AP on both datasets for each set of major basis kernels that satisfied $V_{Q_n} = 0.95$ for each FPN feature map. However, the R_{FM} of the proposed method is less than half compared to case of $W_n \times H_n$, and therefore, the proposed method could be deemed to be more effective for feature map compression as it preserved local and global information within the FPN feature maps across various conditions. In addition, the two points: $V_{Q_n} = 0.85$ for the case of $W_n \times H_n$ and $V_{Q_n} = 0.95$ are performed for the proposed method to compare performance when the amount of compressed data is similar. For $W_n \times H_n$, the AP is approximately 9.85% lower than that for the proposed method. Furthermore, it presented a significant degradation in performance on small objects (AP_S) by approximately 21.12% compared to the original result, which is almost undetectable.

Fig. 6 shows the AP performance according to R_{FM} for each test dataset based on Table 1. Both the PCA-cases and DCT-cases performed the best for the receptive block size

TABLE 1. Performance of the proposed method under several conditions according to the test dataset ($q = 8$).

Transform	Receptive block size	V_{Q_n}	MS COCO 2017 validation set				OpenImageV6	R_{FM} (%)
			AP_L (%)	AP_M (%)	AP_S (%)	AP (%)	AP (%)	
DCT	$W_n \times H_n$	0.80	49.50	22.49	1.81	24.50	59.49	0.27
		0.85	52.84	34.94	6.08	31.01	67.56	0.51
		0.90	54.49	42.87	16.20	38.30	69.47	1.32
		0.95	54.67	44.46	22.76	40.84	72.04	3.21
	$W_5 \times H_5$ [34][35]	0.80	48.26	26.86	2.14	25.17	65.53	0.38
		0.85	51.56	34.88	6.10	31.27	67.83	0.65
		0.90	54.31	42.48	16.62	38.36	70.01	1.14
		0.95	54.25	43.90	22.58	40.70	72.25	2.93
PCA-based transform	$W_n \times H_n$ [21][24][25]	0.80	51.60	30.62	2.96	27.90	62.81	1.86
		0.85	52.71	39.28	4.97	31.55	67.22	2.55
		0.90	53.62	42.21	14.81	37.10	70.69	3.65
		0.95	55.23	43.24	21.27	40.08	72.15	5.72
	$W_5 \times H_5$ (RPCA, Proposed)	0.80	52.22	27.19	2.14	25.17	65.57	0.34
		0.85	54.80	36.83	6.32	32.45	68.98	0.58
		0.90	54.36	44.20	18.46	39.55	71.22	1.11
		0.95	55.15	44.45	22.62	41.40	72.98	2.54
Original (Uncompressed)			55.20	44.23	26.09	45.53	79.08	-

fixed at $W_5 \times H_5$ for each FPN feature map. Moreover, PCA-case outperformed DCT-case in terms of both the AP and compressed data when the receptive block size is $W_5 \times H_5$. Generally, although DCT has less computational complexity than PCA, both are performed for the encoder-side only. From an MPEG-VCM perspective, only decoder-side operations are normative issue. The decoder complexity of the two cases can potentially be almost the same since only inverse transformations, and inverse quantization processes for reconstruction are commonly applied. More specifically, decoder complexity in PCA-case is lower than that of DCT-case since the number of coefficients for PCA-case is generally less than that for DCT-case. Note that the number of coefficients is determined by V_{Q_n}. In addition, decoder complexity of the proposed algorithms is much less than that of video or feature decoders in MPEG-VCM applications, as shown in Figs. 1 (a) and (c). Furthermore, the proposed method is superior from the view of standardization because it can obtain the same detection performance through a smaller amount of data than previous DCT-based methods [34], [35], despite of additional kernels and mean vectors are required to be compressed. Thus, a complexity comparison is performed only for PCA-cases, between extant PCA-based methods [21], [24], [25] and the proposed method. PCA has a complexity in the encoding process of each FPN feature map, and it is derived by referring [43] as shown below:

$$T_n = O\left(R_n \times Z_n \times \min\left(R_n, Z_n\right) + \left(Z_n\right)^3\right), \tag{6}$$

where $R_n = M_n \times N_n$. In both cases of performing the comparison, the complexity of the transform for P_5 is the same, but the computational complexity of transforming the other FPN feature maps is different. For each feature map, R_n is fixed as $W_5 \times H_5$ and Z_n is changed in the proposed method, conversely, R_n is changed and Z_n is fixed as C in the previous PCA-based feature map compression methods [21], [24], [25]. Therefore, the overall encoding computational complexity of the proposed method is slightly higher than that of the extant PCA-based methods [21], [24], [25].

Furthermore, the performance in terms of the quantization bit-depth is performed. As shown in Table 1, most AP matrices are saturated at $V_{Q_n} = 0.90$ for each metric, particularly the AP, AP_M, and AP_S, for the MS COCO 2017 validation set. Therefore, $V_{Q_n} = 0.90$ is chosen as the criterion because feature map compression aims to reduce the volume of compressed data to less than that of input image data without AP degradation.

First, to determine the degradation in AP performance owing to the quantization error, the 32-bit which means without quantization with eight bits, is compared. As indicated in Tables 1 and 2, the resulting performance is not significantly different. Therefore, the proposed uniform integer quantization process did not significantly affect all AP values up to $q = 8$.

Furthermore, even if the quantization bit-depth is adjusted to five, compared to when the bit-depth is eight, R_{FM} could be reduced from 1.11% to 0.70% with subtle degradation of approximately 0.61% and 0.44% in the AP on the MS COCO 2017 validation set and the OpenImageV6 dataset, respectively. By contrast, marginal improvements of approximately 0.53% in AP_S and 0.37% in AP_L on the MS COCO 2017 validation set are observed, which could also be attributed to the quantization error. Therefore,

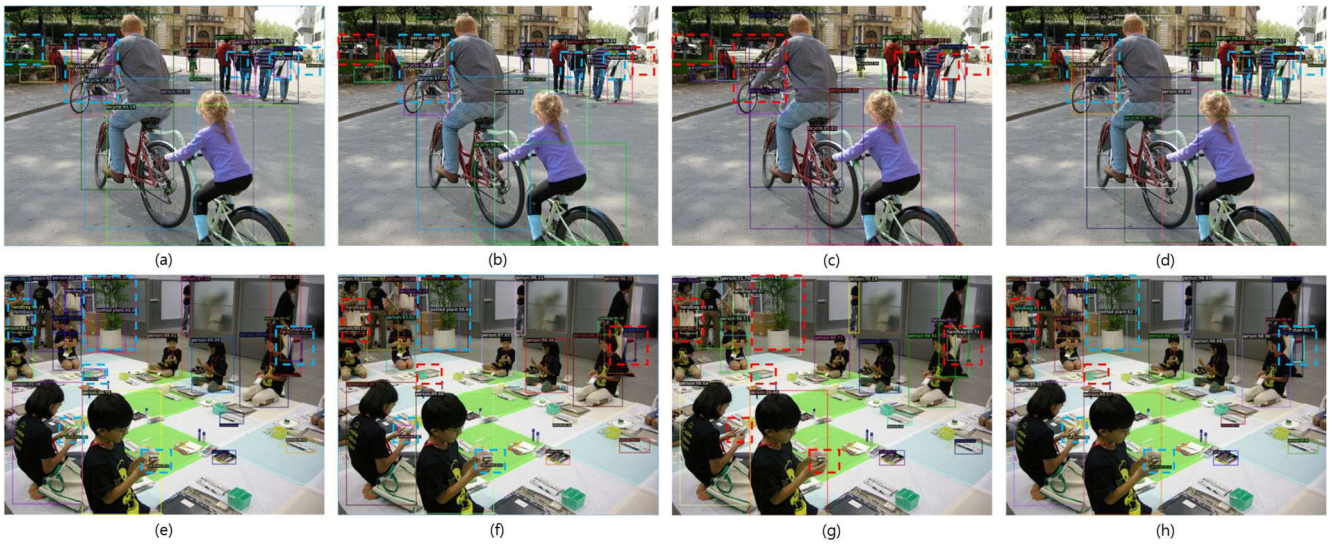

FIGURE 7. Examples of object detection results obtained from the OpenImageV6 dataset. (a), (e) Original (Uncompressed). (b), (f) DCT ($W_5 \times H_5$) [34], [35] with $V_{Q_n} = 0.90$. (c), (g) PCA ($W_n \times H_n$) [21], [24], [25] with $V_{Q_n} = 0.80$. (d), (h) RPCA ($W_5 \times H_5$) (Proposed) with $V_{Q_n} = 0.90$.

TABLE 2. Performance of the proposed method according to quantization bit-depth ($W_5 \times H_5$, PCA-CASE, $V_{Q_n} = 0.90$).

Dataset	q / $AP_{(\%)}$	32-bit (w/o Q)	7	6	5
MS COCO 2017 Val.	AP_L	54.49	54.62	55.31	54.73
	AP_M	44.28	44.37	42.83	43.46
	AP_S	18.52	18.58	18.49	18.99
	AP	39.58	39.47	39.22	38.94
O.ImgV6	AP	71.31	71.13	71.30	70.78
R_{FM} (%)		7.10	0.98	0.73	0.70

as listed in Table 2, the AP is mostly preserved even when the quantization bit-depth is set to five. When uniform integer quantization is performed by setting the quantization bit-depth to four, the AP is significantly deteriorated. Hence, uniform integer quantization is performed for up to five bits in this work.

In addition, the experimental results are visually inspected for certain images from the OpenImageV6 dataset. Fig. 7 displays the object detection results for four cases: Original (Uncompressed), DCT ($W_5 \times H_5$) [34], [35] with $V_{Q_n} = 0.90$, PCA ($W_n \times H_n$) [21], [24], [25] with $V_{Q_n} = 0.80$, and the proposed method. the conditions of each case are selected to compare the performance when the amount of input image and compressed data is similar. The dotted bounding box as shown in Fig.7 highlights a noticeable difference in the object detection results for each case. More specifically, the blue dotted bounding box represent objects that are detected as in the Original (Uncompressed) result, and the red dotted bounding box represent the undetected objects. As shown in Fig. 7, the proposed method presents superior results for the detection of particularly small objects compared to other methods. Moreover, the detection results of the proposed method are more similar to the Original (Uncompressed) results than the other results.

It is concluded that the proposed method outperforms other methods with respect to all AP metrics, such as AP_S, AP_M, AP_L, and AP for a similar volume of compressed data. In particular, the proposed method preserves more information within P_2, used for small-object detection, than the previous PCA-based feature map compression methods [21], [24], [25]. Consequently, the FPN feature maps can be compressed using the proposed method under the condition that the size of the receptive block is $W_5 \times H_5$, with $V_{Q_n} = 0.90$, and the quantization bit-depth $= 5$. Thus, the machine consumption task part can be performed with minimal deterioration, particularly in AP_M, and in AP_L performance by using reconstructed FPN feature maps by performing inverse transform and de-quantization, accounting for only 0.70% of the original FPN feature maps data, which is a smaller percentage than the input image data.

V. CONCLUSION

In this work, a feature map compression method was proposed with a transform process using RPCA and uniform integer quantization. The proposed method could be used to compress feature maps for the output of feature extractor modules in potential MPEG-VCM structures to effectively reduce data in the feature maps. First, the redundant data within each FPN feature map extracted from X101-FPN was eliminated using RPCA. Based on the fact that the size of the receptive block differed according to the channel size of the FPN feature maps extracted from the FPN structure, a transform based on RPCA were performed by rearranging and column-wising the feature map such that the global and local information was preserved. Subsequently, the transformed data were additionally compressed using uniform integer quantization. Further, experimental results revealed that the proposed method reduced the amount of FPN feature maps by 99.30%, with an AP loss of 8.30% on

the OpenImageV6 dataset and AP_L, AP_M, and AP_S losses of 0.47%, 0.77%, and 7.10%, respectively, on the MS COCO 2017 validation set. In addition, the method was found to be superior to the previous feature map compression methods based on PCA and DCT even at higher compression rates.

Furthermore, compared to compression and transmission image/video data methods, the proposed method presented the advantage of transmitting less data than image/video and requiring low processing on the device after decoding the feature map data. However, further studies on efficient compression methods are needed to effectively transmit the feature maps.

REFERENCES

[1] Y. Zhang, S. Kwong, and S. Wang, "Machine learning based video coding optimizations: A survey," *Inf. Sci.*, vol. 506, pp. 395–423, Jan. 2020.
[2] S. Bhattacharya, S. R. K. Somayaji, T. R. Gadekallu, M. Alazab, and P. K. R. Maddikunta, "A review on deep learning for future smart cities," *Internet Technol. Lett.*, vol. 5, no. 1, p. e187, May 2020.
[3] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan. 2018.
[4] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robot.*, vol. 37, no. 3, pp. 362–386, Apr. 2020.
[5] S. P. Yadav, "Vision-based detection, tracking, and classification of vehicles," *IEIE Trans. Smart Process. Comput.*, vol. 9, no. 6, pp. 427–434, Dec. 2020.
[6] G. Sreenu and M. A. S. Durai, "Intelligent video surveillance: A review through deep learning techniques for crowd analysis," *J. Big Data*, vol. 6, pp. 1–27, Jun. 2019.
[7] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[8] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1648–1667, Dec. 2012.
[9] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, "Developments in international video coding standardization after AVC, with an overview of Versatile Video Coding (VVC)," *Proc. IEEE*, vol. 109, no. 9, pp. 1463–1493, Jan. 2021.
[10] M. Rafie, L. Yu, Y. Zhang, and S. Liu, *Call for Evidence for Video Coding for Machines*, document m56229, ISO/IEC JTC 1/SC 29/WG 2 MPEG, Jan. 2021.
[11] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," in *Proc. ACM SIGARCH*, vol. 45, no. 1, 2017, pp. 615–629.
[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[13] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 1492–1500.
[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 2117–2125.
[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Mar. 2017, pp. 2980–2988.
[16] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 1–17.
[17] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6201–6210.
[18] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, and W. Liu, "Deep high-resolution representation learning for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern. Recognit. (CVPR)*, Mar. 2020, pp. 1–23.
[19] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, pp. 611–629, Aug. 2018.
[20] Y. Wang, C. Xu, C. Xu, and D. Tao, "Beyond filters: Compact feature map for portable deep model," in *Proc. 34th Int. Conf. Mach. Learn.*, Aug. 2017, pp. 3703–3711.
[21] E. Son and C. Kim, *CNN Intermediate Feature Coding for Object Detection*, document m54307, ISO/IEC JTC1/SC29/WG11 MPEG2020. Jun. 2020.
[22] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7263–7271.
[23] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[24] B. Chmiel, C. Baskin, E. Zheltonozhskii, R. Banner, Y. Yermolin, A. Karbachevsky, A. M. Bronstein, and A. Mendelson, "Feature map transform coding for energy-efficient CNN inference," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–9.
[25] F. Xiong, F. Tu, M. Shi, Y. Wang, L. Liu, S. Wei, and S. Yin, "STC: Significance-aware transform-based codec framework for external memory access reduction," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6.
[26] 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.
[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[28] H. Choi, *A Result of Feature Data Reduction Using PCA for Object Detection*, document m55414, ISO/IEC JTC1/SC29/WG11 MPEG2020, Oct. 2020.
[29] H. Choi and I. V. Bajić, "Deep feature compression for collaborative object detection," in *Proc. IEEE ICIP*, Oct. 2018, pp. 3743–3747.
[30] Y. Yoon and J. Kim, "Efficient representation of video features for VCM," in *Proc. Korean Soc. Broadcast Eng. Conf.*, Nov. 2020, pp. 183–186.
[31] K. Fischer, F. Brand, C. Herglotz, and A. Kaup, "Video coding for machines with feature-based rate-distortion optimization," in *Proc. IEEE 22nd Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2020, pp. 1–6.
[32] S. Xia, K. Liang, W. Yang, L.-Y. Duan, and J. Liu, "An emerging coding paradigm VCM: A scalable coding approach beyond feature and signal," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2020, pp. 1–6.
[33] S. Yang, Y. Hu, W. Yang, L.-Y. Duan, and J. Liu, "Towards coding for human and machine vision: Scalable face image coding," *IEEE Trans. Multimedia*, vol. 23, pp. 2957–2971, 2021.
[34] S. H. Kim, E.-S. Park, M. Ghulam, and E.-S. Ryu, "Compression method for CNN models using DCT," in *Proc. Korean Soc. Broadcast Eng. Conf.*, Jul. 2020, pp. 553–556.
[35] Z. Shao, X. Chen, L. Du, L. Chen, Y. Du, W. Zhuang, H. Wei, C. Xie, and Z. Wang, "Memory-efficient CNN accelerator based on interlayer feature map compression," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 2, pp. 668–681, Feb. 2022.
[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Represent.*, 2015, pp. 1–14.
[37] R. J. Cintra and F. M. Bayer, "A DCT approximation for image compression," *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 579–582, Oct. 2011.
[38] V. K. Goyal, J. Zhuang, and M. Veiterli, "Transform coding with backward adaptive updates," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1623–1633, Jul. 2000.
[39] C.-Z. Di, C. M. Crainiceanu, B. S. Caffo, and N. M. Punjabi, "Multilevel functional principal component analysis," *Ann. Appl. Statist.*, vol. 3, no. 1, pp. 458–488, Mar. 2009.
[40] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2
[41] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.
[42] G. Ai. (2020). *OpenImage Dataset*. [Online]. Available: https://storage.googleapis.com/openimages/web/index.html
[43] T. Elgamal and M. Hefeeda, "Analysis of PCA algorithms in distributed environments," 2015, *arXiv:1503.05214*.

MINHUN LEE received the B.S. degree in mathematics and electronic engineering (double major) and the M.S. degree in electronic engineering from Kwangwoon University, Seoul, South Korea, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree in computer engineering. His current research interests include video coding, screen content coding, video processing, 3-D reconstruction, and computer vision.

HANSOL CHOI received the B.S. and M.S. degrees in computer engineering from Kwangwoon University, Seoul, South Korea, in 2018 and 2020, respectively, where she is currently pursuing the Ph.D. degree in computer engineering. Her current research interests include video coding, video processing, and computer vision.

JIHOON KIM received the B.S. degree in electronics and communications engineering and the M.S. degree in computer engineering from Kwangwoon University, Seoul, South Korea, in 2019 and 2021, respectively. His current research interests include deep learning, video processing, and computer vision.

JIHOON DO received the B.S. and M.S. degrees in electronics and telecommunication engineering from Korea Aerospace University, South Korea, in 2018 and 2020, respectively. Since 2020, he has been a Researcher with the Electronics and Telecommunications Research Institute (ETRI). His research interests include video coding, feature coding, and machine learning.

HYOUNGJIN KWON received the B.S. and M.S. degrees in electrical and electronic engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1997 and 2001, respectively. Since 2001, he has been with the Electronics and Telecommunications Research Institute (ETRI), where he is currently a Principal Researcher with the Telecommunication and Media Research Laboratory. His research interests include various aspects of image processing and video coding, machine learning-based signal representation and processing, and multimedia telecommunication systems.

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 Korea Advanced Institute of Science and Technology (KAIST), in 2014. In 1996, he joined the Electronics and Telecommunications Research Institute (ETRI), as a Principal Researcher. He has many contributions to the development of international standards, such as scalable video coding and high efficiency video coding. His current research interests include video coding for machines and AI-based video coding.

DONGGYU SIM (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering and the Ph.D. degree from Sogang University, South Korea, in 1993, 1995, and 1999, respectively. He was with Hyundai Electronics Company Ltd., from 1999 to 2000, being involved in MPEG-7 standardization. He was a Senior Research Engineer with Varo Vision Company Ltd., where he was working on MPEG-4 wireless applications, from 2000 to 2002. He was with the Image Computing Systems Laboratory (ICSL), University of Washington, as a Senior Research Engineer, from 2002 to 2005. He researched on ultrasound image analysis and parametric video coding. Since 2005, he has been with the Department of Computer Engineering, Kwangwoon University, Seoul, South Korea. In 2011, he joined Simon Frasier University as a Visiting Scholar. He is one of the leading inventors in many essential patents licensed to MPEG-LA for HEVC standard. His current research interests include video coding, video processing, computer vision, and video communication.

SEOUNG-JUN OH (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1980 and 1982, respectively, and the Ph.D. degree in electrical and computer engineering from Syracuse University, Syracuse, NY, USA, in 1988. In 1988, he joined the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea, as a Senior Research Member, where he was promoted to a Program Manager of Multimedia Research Session. Since 1992, he has been a Professor of the Department of Electronics Engineering, Kwangwoon University, Seoul. He has been the Chairperson of SC29-Korea, since 2001. His research interests include image/video processing, computer vision, real-time video processing, and machine/deep learning for computer vision.