ABCD : Arbitrary Bitwise Coefficient for De-quantization

Woo Kyoung Han Byeonghun Lee Sang Hyun Park Kyong Hwan Jin* Daegu Gyeongbuk Institute of Science and Technology (DGIST), Korea {cjss7894, qudgns1113, shpark13135, kyong.jin}@dgist.ac.kr

Abstract

Modern displays and contents support more than 8 bits image and video. However, bit-starving situations such as compression codecs make low bit-depth (LBD) images (<8 bits), occurring banding and blurry artifacts. Previous bit depth expansion (BDE) methods still produce unsatisfactory high bit-depth (HBD) images. To this end, we propose an implicit neural function with a query to recover de-quantized images from arbitrarily quantized inputs. We develop a phasor estimator to exploit the information of the nearest pixels. Our method shows superior performance against prior BDE methods on natural and animation images. We also demonstrate our model on YouTube UGC datasets for de-banding. Our source code is available at https://github.com/WooKyoungHan/ABCD

1. Introduction

The bit depth in digital contents means a number of binary digits representing pixel values. As humans recognize a wide range of color and luminance, modern display devices and cameras support more than the 8-bit depth of image and video [21, 28]. Regardless of these efforts, the images and video codecs enforce HBD images to be quantized into LBD images due to bit starvation. Thus, most contents are under 8 bits leading to false contours and blurry artifacts. Bit depth expansion, a.k.a. de-quantization, aims to recover missing bits caused by such quantizations.

Conventional methods such as [6, 10, 19, 24, 36–38] have been proposed for the de-quantization problem. However, these methods suffer from blurry artifacts resulting in distortions of details or false contours in extreme BDE. Recently, learning-based approaches, a.k.a. deep neural network, have shown remarkable performances in BDE [4, 9, 18, 26, 32, 40, 43]. Most learning-based approaches [4, 9, 32, 43] reconstruct HBD images in an end-to-end manner. Recent methods [18, 26] recover residual components corresponding to missing bits from LBD images. In particular, the method called D16 [26], with the best performance so far, conducts a binary classification per each bit plane. However, D16 requires multiple deep neural networks models for every bit-planes.

Recently, an implicit neural representation (INR) which maps coordinates into signal values [25, 29], shows promising performances in various tasks [5, 11, 15, 22, 25, 29]. The implicit neural networks have a spectral bias problem toward low frequencies, which makes INR hard to represent high-frequency components [27]. Fortunately, several solutions are developed to relax the spectral bias [15, 23, 30, 33, 41]. However, there is no INR approach for bit depth expansion problems.

In this paper, we propose a novel model, the Arbitrary Bit-wise Coefficient model for De-quantization (ABCD), to recover missing bits from the randomly quantized LBD image to any HBD image. The proposed model addresses the spectral bias of INR and improves de-quantization quality through the use of an encoder to estimate the dominant phasors in the ground truth. As shown in Fig. 1, our encoder estimates the dominant phasors to mitigate the spectral bias of INR. Then, the model utilizes an INR to achieve arbitrary-bit reconstructions in the amplitude domain. Finally, a bit
decoding step converts bit coefficients into HBD images by multiplying the bit-basis. The proposed model represents a significant advancement over previous de-quantization techniques with providing high flexibility and accuracy as it effectively recovers missing bits from randomly quantized inputs.

In summary, our main contributions are as follows:

• We propose a bit depth expansion algorithm using an implicit neural representation with a bit query in arbitrarily quantized bit levels and demonstrate our method achieves state-of-the-art performance.

• We show that the proposed phasor estimator predicts the dominant phasors of the ground truth coefficient’ in the Fourier domain.

• We validate our pre-trained model not only on five image datasets as de-quantization but also on the YouTube-UGC dataset as de-banding.

2. Related Work

Bit depth expansion There are straightforward ways for BDE, such as zero padding (ZP) method and the bit replication (BR) [36], which sets ‘0’ or most significant bits (MSBs) for missing bits. Even though these algorithms are hardware friendly, each reconstructed signal depends only on its value without considering surrounding pixels. In contrast, interpolation methods [6, 37], content-adaptive (CA) BDE, and contour region reconstruction (CRR) effectively remove false contour artifacts. However, they blur the details. Intensity potential for adaptive de-quantization (IPAD) method [19] proposed an iterative algorithm which uses the intensity potential field calculated with the connected component label. IPAD achieved higher PSNR than the algorithms above. However, IPAD still suffers from false contour artifacts in large BDE.

Meanwhile, the BDE is highly related to de-banding. The banding artifacts are staircase-like color phenomena of images, especially in video contents. Previous works [31, 34] resolved de-banding with adaptive filter-based methods. Because the artifact is mainly derived from compression with quantization, we apply our de-quantization method to remove the banding artifact.

Learning-based Bit Recovery The proposed neural network algorithms [4, 18, 32, 40] predict favorable HBD images with higher performance than the aforementioned expert rule systems [19, 37]. The BE-CALF [18] performs BDE by recovering residuals of LBD images. However, it only supports dedicated bit LBD images as inputs. Even though the BitNet [4] supports 3 to 6-bit inputs with a single model, the performance is not as good as BE-CALF. Unlike previous works, the D4 and D16 networks [26] employ the bit-wise classification per each missing bit-plane. However, multiple models are required for predicting each bit-plane (e.g., 4 to 8-bit BDE requires four independent models). Furthermore, Fig. 2 shows that the learning-based approach [26] is not able to remove false contours completely. Unlike the prior works, we designed the network receiving randomly quantized LBD images with single training.

Implicit Neural Representation (INR) Recently, various tasks [11, 22, 29] apply a multi-layer perceptron (MLP) as implicit neural representation. Although the INR parameterizes coordinates to continuous signals with memory-efficient frameworks, INR has two issues: per-scene optimization and spectral bias. To overcome a problem of per-scene optimization, prior works [5, 15] concatenate latent features from inputs with coordinates. The spectral bias [27] induces the network to learn mainly low frequencies. Recently, in the super-resolution task, the local texture estimator (LTE) [15] overcomes such spectral bias by learning high-resolution images’ dominant frequency and phases. Nevertheless, there is no INR for de-quantization to our best knowledge. Therefore, we suggest INR as a function of given LBD data and amplitude coordinates calculated with input and output bit-depth. Moreover, we develop
a phasor estimator so that the network avoids spectral bias.

3. Problem Formulation

In this section, we analyze images with modular arithmetic properties and formulate implicit neural representation as a function of amplitude. Let \( I_N \) and \( I_q \) be an HBD image and an LBD image, respectively, with bit-plane zero padding when \( q < N \). Then, the quantization equation from \( I_N \) to \( I_q \) is defined as below:

\[
I_q = \left\lfloor \frac{I_N}{2^{N-q}} \right\rfloor 2^{N-q}, \tag{1}
\]

where \( \lfloor \cdot \rfloor \) is a floor function mapped to the greatest integer. \( I_q \) contains \( q \) MSBs of \( I_N \). Then, The HBD image is a sum of the LBD image \( I_q \) and the residual image \( R \) as below:

\[
I_N = I_q + R. \tag{2}
\]

**Decomposition of Quantized Signal** Our approach estimates the residual image \( R \) with a function of both encoded latent variables and a bit-wise query. Here, we define bit-wise basis and its coefficients which normalize quantization residuals. The binary set \( \{0, 1\} \) together with exclusive-or (XOR) operation \( \odot \) and multiplication \( \cdot \) compose a binary field denoted by \( F_2 \). It is known that binary vector space \( \{0, 1\}^N \) over \( F_2 \) have orthonormal basis as one-hot encoding denoted as \( e_n \) where \( n = 0, \ldots, N-1 \) and \( n \) is integer [1]. We interpret an arbitrary \( N \)-bit number as an element of the \( N \)-dimensional binary vector space \( (\{0, 1\}^N, \odot, \cdot). \)

To realize such a binary vector in a real number, we present an arbitrary positive number \( a \) as a power series of 2 with elements \( (b_i) \) from a binary vector:

\[
a = \sum_{i=-\infty}^{\infty} b_i 2^i, \tag{3}
\]

where \( b_i \in \{0, 1\} \) and \( i \in \mathbb{Z} \). When we split above series in Eq. (3) into two terms with respect to any integer \( L \),

\[
a = \sum_{i=L+1}^{\infty} b_i 2^i + \sum_{j=-\infty}^{L} b_j 2^j = \sum_{i=L+1}^{\infty} b_i 2^i + C \cdot 2^{L+1}, \tag{4}
\]

where \( C \in [0, 1] \). Note that digital images consist of positive integers such as “uint8”. We present Eq.(2) with Eq.(4) into a digital image form as below:

\[
I_N = \sum_{i=N-q}^{N-1} 2^i \cdot B_i + 2^{N-q} \cdot C_i. \tag{5}
\]

where \( B_i \in \{0, 1\}^{H \times W \times 3} \) denotes a binary image at \( i \)-th least significant bit-plane. Thus, the residual image \( R \) is a multiplication between \( 2^{N-q} \) and the real number image \( C \in [0, 1]^{H \times W \times 3} \). The orthonormal basis \( e_q \) of the binary vector space is equivalent to \( 2^{N-q} \) in the arithmetic form. From now on, we call \( C \) bit-wise coefficient of the bit-wise query \( (s := 2^{N-q}) \) which is equivalent to \( e_q \).

**Arbitrary Bitwise Coefficients** Our method, ABCD, aims to predict coefficient \( C \) to reconstruct residual images by using the implicit neural representation (INR) as a function of latent vector and its quantizing bit-query. The INR parameterizes a continuous signal with an MLP fed by coordinates \([5, 23, 29]\). We design INR \( f_\theta \) using amplitudes as an input coordinate so that it represents bit-wise coefficients \( C \) as continuous signals along the amplitude axis as follows:

\[
C(x, s) \simeq f_\theta(I_q[N(x)], s). \tag{6}
\]

where \( x \) is a 2D coordinate in the image domain and \( N(x) \) is a set of nearest pixels around the center pixel, \( x \). A decoder \( f_\theta \) maps coefficient image \( \hat{C} \) from domain composed with latent vector and bit-wise query; \( f_\theta(z, s) : (Z, S) \mapsto \hat{C} \), where \( z \in Z \) is a latent tensor from an encoder \( E_\varphi, S \)
is set of basis \( e_i \) and \( C \) is a space of predicted coefficient values from \( f_\theta \).

**Phasor Estimator** Inspired from [15, 33], we insert phasor estimator by modifying local texture estimator [15] to relax spectral bias. We demonstrate the difference between our ABCD and local texture estimator in Sec. 6. The overall system is constructed as follows:

\[
\hat{C}(x, I_q, s; \Theta) = f_\theta(h_\psi(z_x), s),
\]

where \( z = E_\varphi(I_q) \), \( h_\psi(\cdot) \) denotes the phasor estimator of ABCD and \( \Theta = \{\theta, \varphi, \psi\} \) is a set of trainable parameters. Our phasor estimator \( (h_\psi(\cdot)) \) consists of two elements: (1) an amplitude estimator \( (h_a(\cdot) : R^C \rightarrow R^{2K}) \) and (2) a phase estimator \( (h_p(\cdot) : R^C \rightarrow R^{2K}) \). Thus, given a coordinate \( x \in R^2 \), the estimating function \( h_\psi(\cdot) : (R^C) \rightarrow R^{2K} \) is defined as

\[
h_\psi(z_x) = \left[ \begin{array}{c} A^1_x \\ A^2_x \end{array} \right] \circ \left[ \begin{array}{c} \cos(\pi \Phi^1_x) \\ \sin(\pi \Phi^1_x) \end{array} \right],
\]

where \( A^1_x, A^2_x \) = \( h_a(z_x) \), \( \Phi^1_x, \Phi^2_x \) \( = h_p(z_x) \).

**Bit decoding** To retrieve de-quantized image values as in Eq. (1), bitwise coefficients \( C \) is multiplied with basis value \( s \) and added with corresponding input value \( I_q \) from bitplane zero-padding. The whole process of decoding is written as follows:

\[
\hat{I}_N(x) = \hat{C}(x, I_q, s; \hat{\Theta}) \times s + I_q(x)
\]

**4. Method**

**4.1. Network Detail**

Our ABCD-based arbitrary BDE network includes an encoder \((E_\varphi)\), a phasor detector (blue shaded area in Fig. 3), a decoder \((f_\theta)\), and a bit-wise coefficient estimator (pink shaded area in Fig. 3). This section describes a backbone structure (including encoder and decoder) and architectural details of ABCD.

**Encoder \((E_\varphi)\) and Decoder \((f_\theta)\)** We use EDSR [17], RDN [42], and SwinIR [16] as an encoder \((E_\varphi)\). The EDSR [17] is composed of 38 ResBlocks with 128 channels. The RDN [42] is composed of 8 RDB blocks. Since the BDE task requires equivalent sizes between input and output, we apply encoders without upsampling layers. The decoder \((f_\theta)\) is composed with 5-layer MLP and ReLU activation function \([5, 15]\). The dimension of the first layer is 257, and the hidden dimensions are 256.

**ABCD** Our ABCD contains a phasor estimator and a bit-wise coefficient estimator. Inspired by the LTE [14, 15], we hypothesize that our ABCD learns the phasor distribution from Fourier representations. The phasor estimator contains amplitude estimator \( (h_a) \), phase estimator \( (h_p) \) and sinusoidal activations. These estimators consist of \( 3 \times 3 \) convolution layers having 256 output channels. Bit-wise coefficient estimator calculates bit-query \( s \) following Eq. (5), and concatenates it with the output of the phasor estimator \((\hat{E}_\psi)\). The concatenated vector contains 256 channels which are identical to an input dimension of MLP. We clip the output of network with normalized tanh activation \((0 \leq \hat{C} \leq 1)\) to prevent overshot and undershoot effect. Our network reconstructs HBD images by multiplying coefficients \( C \) with bit-query \( s \) as in Eq. (11).

**4.2. Training Strategy**

We construct a minibatch with uniformly sampled quantization levels from 3-bit to 8-bit. Note that our network shows robustness for unseen levels (2-bit or 10-bit). Let \( q \) be a quantization level randomly sampled from 3 to 8 integer. With Eq. (1), we quantize the HBD image to \( q \)-bit and add bit-wise zero padding before feeding into a network. We calculate the bit-wise coordinate \( s = 2^{N-q} \cdot \lfloor N/16 \rfloor \) and divide \( R \) by \( s \). Randomly sample pixels from ground truth (GT) \((C)\).

**5. Experiment**

**5.1. Training**

**Dataset** As in [26], we use 2000 16-bit images, each 1000 from the Sintel dataset and the MIT-Adobe FiveK dataset [3] for training. For evaluation, we report peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) on the MIT-Adobe FiveK [3], Sintel datasets [7], benchmarks for TESTIMAGES 1200 [2], Kodak [8], and ESPL v2 [13].

**Implementation detail** We use \( 64 \times 64 \) patches for inputs of our network and optimize it by Adam [12]. We use 1000 epochs with a batch size 16. When we train ABCD with CNN-based encoders, such as EDSR [17] or RDN [42], the learning rate is initialized to 1e-4 and decayed by factor of 0.5 at [200, 400, 600, 800]. For a transformer encoder (SwinIR) [16], the learning rate is initialized as 1e-5 and decayed by factor of 0.5 at [500, 800, 900, 950].
5.2. Evaluation

Qualitative result We use 8-bit ground truth images since the standard displays support up to 8-bit. Fig. 4 shows a qualitative comparison to other BDE methods, IPAD [19], BitNet [4], and D16 [26]. We found that D16 [26] suffers from false contour and BitNet [4] blurs details. The first and second rows demonstrate that our ABCD has the advantage of reconstructing details blurred by quantization. Also, overall comparisons show that our ABCD is effective in removing false contour artifacts. Furthermore, in Fig. 5, we demonstrate that our ABCD restores the extreme bitplanes. Note that 2-bit inputs in Fig. 5 are unused samples for training.

Quantitative result We compare the performance of our method against existing methods: IPAD [19], which is a non-learnable method, BitNet [4], BE-CALF [18], and D16 [26]. The input in Tab. 1 refers to $q$-quantized images with zero-padding at missing bitplanes. The number of trainable parameters is written below. The pre-trained BitNet supports 3-bit or 4-bit to 8-bit and 3 to 6-bit to 16-bit expansion. The BE-CALF [18] provides a pre-trained model for 4-bit or 8-bit to 16-bit expansion; however, the training code is not available. We directly copy the numeric results from the original papers. In Tab. 1 the test dataset is composed of randomly selected 50 images in the Sintel dataset and the last 1000 (filenames a4001 to 5000) images of the MIT-Adobe dataset enhanced by expert E. Our ABCD outperforms all methods with any encoder. The maximum gain of PSNR is 1.52dB on Sintel for 4→16 BDE.

Table 1. Quantitative comparisons (PSNR (dB) & SSIM) for arbitrary bit-depth expansion on the test set of Sintel [7] & MIT-Adobe FiveK [3](top) and benchmark set including TESTIMAGES 1200 [2], Kodak [8], ESPL v2 [13] (bottom). Red and blue colors indicate the best and the second-best performance, respectively. (-) indicates not reported. (q → N) refers q-bit input to N-bit output BDE.

Figure 5. Qualitative comparison for 2-bit → 8-bit BDE. D16 [26] does not provide 2-bit recovery pre-trained model. SwinIR [20] is used as an encoder for ABCD.

other methods. Our ground truth of training is 16-bit depth images, so 8 and 12-bit target expansion are out-of-range for ABCD. However, thanks to the INR that maps bit-wise coordinates to its coefficient, ABCD achieves the highest PSNR.

5.3. Ablation Study

Network components In Tab. 2, we conducted ablation studies for individual components of ABCD. ABCD contains a phasor estimator in the encoder and bit-wise coordinate concatenation in the decoder. Furthermore, our output is the bit-wise coefficient (C) instead of pixel values (I_{HBD} or R). To support this, we train EDSR [17]-based ABCD without each component and mechanism. (-P): Encoder without phasor estimator (replaced by ResBlocks), (-S): Decoder without bit-wise query (s), (+B): Mechanism that estimates residual image (R), and (+L): Addition of long skip connection of LBD images (I_{LBD}) so that the network predicts the natural image (I_{HBD}).

The middle part (-P, -S) of Tab. 2 shows that the phasor estimator and bit-wise query enhance the performance of ABCD, especially in relatively low bit-depth input. The phasor estimator improved performance in a high range. Without the bit-wise query, it makes the output bit-depth to be dedicated (16-bit) and causes performance drops in the network. The ablations of mechanisms ABCD(+B) and ABCD(+L) degrade the performances of the network. For extracting the residual image (R), arbitrary residual images
Table 2. **Quantitative ablation study** of ABCD on TESTIMAGES 1200 [2] (PSNR(dB)). Definitions of -P, -S, +B, +L are shown in Sec. 5.3. EDSR [17] is used as an encoder.

<table>
<thead>
<tr>
<th>Input Bit</th>
<th>Output Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 → 16</td>
<td>47.67</td>
</tr>
<tr>
<td>4 → 16</td>
<td>52.78</td>
</tr>
<tr>
<td>5 → 16</td>
<td>57.24</td>
</tr>
<tr>
<td>6 → 16</td>
<td>61.78</td>
</tr>
<tr>
<td>7 → 16</td>
<td></td>
</tr>
<tr>
<td>8 → 16</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Ablation study of ABCD for a dedicated bit-depth expansion. Evaluated on TESTIMAGES 1200 (PSNR(dB)). Only-k refers to training the model with k-bit quantized images.

<table>
<thead>
<tr>
<th>Input Bit</th>
<th>Output Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 → 16</td>
<td>41.65</td>
</tr>
<tr>
<td>4 → 16</td>
<td>52.78</td>
</tr>
<tr>
<td>6 → 16</td>
<td>61.78</td>
</tr>
<tr>
<td>8 → 16</td>
<td></td>
</tr>
<tr>
<td>10 → 16</td>
<td></td>
</tr>
</tbody>
</table>

5.4. Phasor Estimation

We demonstrate that our phasor estimator extracts dominant phasors from quantized inputs. For sanity check, we sort the phasor of ground truth ($C$) in descending order in the absolute value of amplitude and select $K$ phasors. With the formula below, we calculate estimated phases, from Eq. (8):

$$\angle \Phi_x = \tan^{-1} \left( \frac{A_x \cos (\pi \Phi_x) + A_x \sin (\pi \Phi_x)}{A_x \sin (\pi \Phi_x) + A_x \cos (\pi \Phi_x)} \right)$$  \hspace{1cm} (12)

We compared the distribution of phasors between GT and the predicted in low-frequency and high-frequency textures of quantized inputs. In Fig. 6, red boxes in the first column are the receptive field (RF) of ABCD, and the second column refers the ground truth of $\hat{C}$. The diagram in Fig. 6 represents an accumulated number for each angle. Note that the Fourier transform of images is conjugate symmetric. We find that our phasor estimator learns the distribution of the dominant phasors of both high-frequency and low-frequency textures. For example, in the third row of Fig. 6, the density of dominant phasors is accumulated near $\pi$ (rad), and our ABCD follows those phasors.

5.5. Debanding

We validate the effectiveness of our ABCD for debanding in unseen dataset. Inspired by the relations between false contour artifacts and quantization, we hypothesize that our pre-trained ABCD can resolve banding effects. We calculate the blind banding artifact detector (BBAND) score [35] between original and enhanced frames by ABCD. As we have no information about how many bits are required for ABCD to resolve banding effects, we assume that severe artifacts demand large bit depth expansion. We enhance original images with $3 \rightarrow 8$ BDE results ($\hat{C}_{3 \rightarrow 8}$). The quan-
Figure 8. **False contour artifacts and analysis in bitplanes** for 3-bit→8-bit bit depth expansion. Comparison with D16 [26], EDSR-ABCD and SwinIR-ABCD.

6. Discussion

**Phasor Estimator** The dependency on the input image sets the phases of LTE [15] and ABCD apart from each other. While LTE infers phases from their scale factor \((c := (2/r_x, 2/r_y))\), ABCD estimates phases from images. Since LTE is a super-resolution network, it requires local coordinates \((\delta := x_{\text{query}} - x_{\text{nearest}})\) to learn frequencies. Unlike LTE, outputs of ABCD always have the same resolution with the input images. This implies that local coordinates \(\delta\) are \(\vec{0}\). Thus, the formulation is given as follows:

\[
A_j \circ \begin{bmatrix} \cos(\pi \Phi_j \delta + h_j(\hat{e})) \\ \sin(\pi \Phi_j \delta + h_j(\hat{e})) \end{bmatrix} \delta = \vec{0} \rightarrow \begin{bmatrix} A_1^j \\ A_2^j \end{bmatrix} \circ \begin{bmatrix} \cos(\pi \Phi_j^1 \delta) \\ \sin(\pi \Phi_j^1 \delta) \end{bmatrix}
\]

**Artifacts from Encoder** In Fig. 8, CNN-based BDE methods have false contour artifacts in restoring a high range of bit-depth. This artifact appears when networks restore high pixel values nearby low-valued pixels. Red arrows indicate such artifacts. Since contour artifacts do not appear on the GT as well as the 3-bit input, it implies that artifacts are caused by the network. We decompose each of the predictions in bit-planes to see behaviors in bit-planes. We overcome these artifacts by using the attention model, SwinIR [16].

**FLOPs and Memory** In Tab. 5 and Fig. 9, we report the number of training parameters, floating point operations (FLOPs), the memory consumption and the average computation time. We test 4 to 16 bit-depth expansion in TESTIMAGES1200 [2] datasets. D16 [26] requires such parameters because of per-bit processing. For a fair comparison, all methods run on Google Colab. Although the BitNet [4] and BE-CALF [18] have smaller sizes and FLOPs than ours, their results are about 3dB lower than ours. We apply EDSR-baseline [17] and SwinIR-lightweight [16] from their official code and confirm that our framework overcomes the trade-off between computational complexities and performances.

7. Conclusion

We proposed an implicit neural network approach as a function of a bit-wise query for BDE. The residual image calculated from bit-wise coefficients recovers the arbitrary depth of missing bit planes with single training. Furthermore, we show that the proposed method removes severe artifacts such as false contour, and blurry artifacts, effectively. Our phasor estimator shows similar phasor diagrams with that of the original image leading to accurate predictions of bit-wise coefficient. The results of test and benchmark datasets demonstrate that our network outperforms state-of-the-art models up to 1.52dB.

Acknowledgement This work was partly supported by the grants of the DGIST R&D program of the Ministry of Science and ICT of KOREA (22-KUJoint-02, 21-JRP-01), Smart HealthCare Program(www.kipot.or.kr) funded by the Korean National Police Agency(KNPA) (No. 230222M01), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub).
References


