



(51) International Patent Classification:

H04N 19/82 (2014.01)

(21) International Application Number:

PCT/CN2024/095040

(22) International Filing Date:

23 May 2024 (23.05.2024)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2023/095950

24 May 2023 (24.05.2023)

CN

(71) Applicants: **DOUYIN VISION CO., LTD.** [CN/CN]; Room B-0035, 2/F, No.3 Building, No.30, Shixing Road, Shijingshan District, Beijing 100041 (CN). **BYTEDANCE INC.** [US/US]; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) Inventors: **LI, Junru**; Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN). **LIN, Chaoyi**; Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN). **LI, Yue**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Kai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG,**

Li; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(74) Agent: **SHIHUI PARTNERS**; 42/F, Tower C, Beijing Yintai Centre, No.2 Jianguomenwai Avenue, Chaoyang, Beijing 100022 (CN).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: METHOD, APPARATUS, AND MEDIUM FOR VIDEO PROCESSING

2600

PERFORM A CONVERSION BETWEEN A CURRENT VIDEO UNIT OF A VIDEO AND A BITSTREAM OF THE VIDEO, WHEREIN THE BITSTREAM COMPRISES AT LEAST ONE INDICATION INDICATING FIRST INFORMATION REGARDING A USAGE OF AN NN-BASED FILTER FOR THE CURRENT VIDEO UNIT, AND THE NN-BASED FILTER IS CONFIGURED TO PROCESS A LUMA COMPONENT AND A CHROMA COMPONENT OF THE CURRENT VIDEO UNIT

2602

Fig. 26

(57) Abstract: Embodiments of the present disclosure provide a solution for video processing. A method for video processing is proposed. The method comprises: performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream comprises at least one indication indicating first information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit.

Published:

— *with international search report (Art. 21(3))*

METHOD, APPARATUS, AND MEDIUM FOR VIDEO PROCESSING

FIELDS

[0001] Embodiments of the present disclosure relates generally to video processing
5 techniques, and more particularly, to neural network-based video coding.

BACKGROUND

[0002] In nowadays, digital video capabilities are being applied in various aspects of
peoples' lives. Multiple types of video compression technologies, such as MPEG-2,
MPEG-4, ITU-TH.263, ITU-TH.264/MPEG-4 Part 10 Advanced Video Coding (AVC),
10 ITU-TH.265 high efficiency video coding (HEVC) standard, versatile video coding (VVC)
standard, have been proposed for video encoding/decoding. However, coding efficiency
of video coding techniques is generally expected to be further improved.

SUMMARY

[0003] Embodiments of the present disclosure provide a solution for video processing.

15 [0004] In a first aspect, a method for video processing is proposed. The method
comprises: performing a conversion between a current video unit of a video and a
bitstream of the video, wherein the bitstream comprises at least one indication indicating
first information regarding a usage of a neural network (NN) based filter for the current
video unit, and the NN-based filter is configured to process a luma component and a
20 chroma component of the current video unit.

[0005] Based on the method in accordance with the first aspect of the present disclosure,
the NN-based filter is capable of processing both luma and chroma components and
information regarding the usage of this NN-based filter is signaled. Compared with the
conventional solution, the proposed method can advantageously enable flexible usage of
25 the NN-based filter. Thereby, the coding efficiency can be improved.

[0006] In a second aspect, an apparatus for video processing is proposed. The apparatus
comprises a processor and a non-transitory memory with instructions thereon. The
instructions upon execution by the processor, cause the processor to perform a method in
accordance with the first aspect of the present disclosure.

[0007] In a third aspect, a non-transitory computer-readable storage medium is proposed. The non-transitory computer-readable storage medium stores instructions that cause a processor to perform a method in accordance with the first aspect of the present disclosure.

5 [0008] In a fourth aspect, another non-transitory computer-readable recording medium is proposed. The non-transitory computer-readable recording medium stores a bitstream of a video which is generated by a method performed by an apparatus for video processing. The method comprises: performing a conversion between a current video unit of the video and the bitstream, wherein the bitstream comprises at least one indication indicating first
10 information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit.

[0009] In a fifth aspect, a method for storing a bitstream of a video is proposed. The method comprises: performing a conversion between a current video unit of the video and
15 the bitstream, wherein the bitstream comprises at least one indication indicating first information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit; and storing the bitstream in a non-transitory computer-readable recording medium.

20 [0010] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

25 [0011] Through the following detailed description with reference to the accompanying drawings, the above and other objectives, features, and advantages of example embodiments of the present disclosure will become more apparent. In the example embodiments of the present disclosure, the same reference numerals usually refer to the same components.

30 [0012] Fig. 1 illustrates a block diagram that illustrates an example video coding system, in accordance with some embodiments of the present disclosure;

- [0013] Fig. 2 illustrates a block diagram that illustrates a first example video encoder, in accordance with some embodiments of the present disclosure;
- [0014] Fig. 3 illustrates a block diagram that illustrates an example video decoder, in accordance with some embodiments of the present disclosure;
- 5 [0015] Fig. 4A shows an example of raster-scan slice partitioning of a picture;
- [0016] Fig. 4B shows an example of a rectangular slice partitioning of a picture;
- [0017] Fig. 4C shows an example of a picture partitioned into tiles, bricks, and rectangular slices;
- [0018] Fig. 4D shows an example of CTBs crossing the bottom picture border;
- 10 [0019] Fig. 4E shows an example of CTBs crossing the right picture border;
- [0020] Fig. 4F shows an example of CTBs crossing the right bottom picture border;
- [0021] Fig. 5 shows an example of encoder block diagram;
- [0022] Fig. 6 illustrates an example of pre-processing and post-processing units;
- [0023] Fig. 7 illustrates an example architecture of the CNN in filter set 0;
- 15 [0024] Fig. 8 illustrates an example implementation of the CNN in filter set 0;
- [0025] Fig. 9 illustrates an example encoder optimization;
- [0026] Fig. 10A illustrates an example head of luma network;
- [0027] Fig. 10B illustrates an example subnetwork;
- [0028] Fig. 10C illustrates another example subnetwork;
- 20 [0029] Fig. 11 illustrates an example temporal in-loop filter;
- [0030] Fig. 12A illustrates an example parameter selection at an encoder side;
- [0031] Fig. 12B illustrates an example parameter selection at a decoder side;
- [0032] Fig. 13 illustrates prediction of a current block from a context of reference samples around the current block via the neural network-based intra prediction mode;
- 25 [0033] Fig. 14 illustrates decomposition of a context of reference samples surrounding the current block into the available reference samples and the unavailable reference

samples;

[0034] Fig. 15 illustrates intra prediction mode signaling for the current luma CB framed in the dashed line;

[0035] Fig. 16 illustrates an example multi-scale neural network with two branches for use in a NN-based loop filter;

[0036] Fig. 17 illustrates an example network structure of NN filter in accordance with some embodiments of the present disclosure;

[0037] Fig. 18 illustrates an example network structure of NN filter in accordance with some embodiments of the present disclosure;

10 **[0038]** Fig. 19 illustrates an example network structure of NN filter in accordance with some embodiments of the present disclosure;

[0039] Fig. 20 illustrates an example network structure of NN filter in accordance with some embodiments of the present disclosure;

15 **[0040]** Fig. 21 illustrates an example network structure of NN filter in accordance with some embodiments of the present disclosure;

[0041] Fig. 22 illustrates an example network structure of NN filter in accordance with some embodiments of the present disclosure;

[0042] Figs. 23A-23C illustrates an example structure and examples parameters of a luma model in accordance with some embodiments of the present disclosure;

20 **[0043]** Figs. 24A-24C illustrates an example structure and examples parameters of a chroma model in accordance with some embodiments of the present disclosure;

[0044] Figs. 25A-25C illustrates an example structure and examples parameters of a single model for processing luma and chroma components in accordance with some embodiments of the present disclosure;

25 **[0045]** Fig. 26 illustrates a flowchart of a method for video processing in accordance with embodiments of the present disclosure; and

[0046] Fig. 27 illustrates a block diagram of a computing device in which various embodiments of the present disclosure can be implemented.

[0047] Throughout the drawings, the same or similar reference numerals usually refer to the same or similar elements.

DETAILED DESCRIPTION

[0048] Principle of the present disclosure will now be described with reference to some
5 embodiments. It is to be understood that these embodiments are described only for the purpose of illustration and help those skilled in the art to understand and implement the present disclosure, without suggesting any limitation as to the scope of the disclosure. The disclosure described herein can be implemented in various manners other than the ones described below.

10 [0049] In the following description and claims, unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skills in the art to which this disclosure belongs.

[0050] References in the present disclosure to “one embodiment,” “an embodiment,”
“an example embodiment,” and the like indicate that the embodiment described may
15 include a particular feature, structure, or characteristic, but it is not necessary that every embodiment includes the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an example embodiment, it is submitted that it is within the knowledge of one skilled in the art to
20 affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0051] It shall be understood that although the terms “first” and “second” etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example,
25 a first element could be termed a second element, and similarly, a second element could be termed a first element, without departing from the scope of example embodiments. As used herein, the term “and/or” includes any and all combinations of one or more of the listed terms.

[0052] The terminology used herein is for the purpose of describing particular
30 embodiments only and is not intended to be limiting of example embodiments. As used herein, the singular forms “a”, “an” and “the” are intended to include the plural forms as

well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises”, “comprising”, “has”, “having”, “includes” and/or “including”, when used herein, specify the presence of stated features, elements, and/or components etc., but do not preclude the presence or addition of one or more other features, elements, components and/ or combinations thereof.

Example Environment

[0053] Fig. 1 is a block diagram that illustrates an example video coding system 100 that may utilize the techniques of this disclosure. As shown, the video coding system 100 may include a source device 110 and a destination device 120. The source device 110 can be also referred to as a video encoding device, and the destination device 120 can be also referred to as a video decoding device. In operation, the source device 110 can be configured to generate encoded video data and the destination device 120 can be configured to decode the encoded video data generated by the source device 110. The source device 110 may include a video source 112, a video encoder 114, and an input/output (I/O) interface 116.

[0054] The video source 112 may include a source such as a video capture device. Examples of the video capture device include, but are not limited to, an interface to receive video data from a video content provider, a computer graphics system for generating video data, and/or a combination thereof.

[0055] The video data may comprise one or more pictures. The video encoder 114 encodes the video data from the video source 112 to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. The I/O interface 116 may include a modulator/demodulator and/or a transmitter. The encoded video data may be transmitted directly to destination device 120 via the I/O interface 116 through the network 130A. The encoded video data may also be stored onto a storage medium/server 130B for access by destination device 120.

[0056] The destination device 120 may include an I/O interface 126, a video decoder 124, and a display device 122. The I/O interface 126 may include a receiver and/or a

modem. The I/O interface 126 may acquire encoded video data from the source device 110 or the storage medium/server 130B. The video decoder 124 may decode the encoded video data. The display device 122 may display the decoded video data to a user. The display device 122 may be integrated with the destination device 120, or may be external to the destination device 120 which is configured to interface with an external display device.

[0057] The video encoder 114 and the video decoder 124 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding (VVC) standard and other current and/or further standards.

10 **[0058]** Fig. 2 is a block diagram illustrating an example of a video encoder 200, which may be an example of the video encoder 114 in the system 100 illustrated in Fig. 1, in accordance with some embodiments of the present disclosure.

[0059] The video encoder 200 may be configured to implement any or all of the techniques of this disclosure. In the example of Fig. 2, the video encoder 200 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video encoder 200. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0060] In some embodiments, the video encoder 200 may include a partition unit 201, a predication unit 202 which may include a mode select unit 203, a motion estimation unit 204, a motion compensation unit 205 and an intra-prediction unit 206, a residual generation unit 207, a transform unit 208, a quantization unit 209, an inverse quantization unit 210, an inverse transform unit 211, a reconstruction unit 212, a buffer 213, and an entropy encoding unit 214.

25 **[0061]** In other examples, the video encoder 200 may include more, fewer, or different functional components. In an example, the predication unit 202 may include an intra block copy (IBC) unit. The IBC unit may perform predication in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[0062] Furthermore, although some components, such as the motion estimation unit 204 and the motion compensation unit 205, may be integrated, but are represented in the example of Fig. 2 separately for purposes of explanation.

[0063] The partition unit 201 may partition a picture into one or more video blocks. The video encoder 200 and the video decoder 300 may support various video block sizes.

[0064] The mode select unit 203 may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra-coded or inter-coded block to a residual generation unit 207 to generate residual block data and to a reconstruction unit 212 to reconstruct the encoded block for use as a reference picture. In some examples, the mode select unit 203 may select a combination of intra and inter predication (CIIP) mode in which the predication is based on an inter predication signal and an intra predication signal. The mode select unit 203 may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter-predication.

[0065] To perform inter prediction on a current video block, the motion estimation unit 204 may generate motion information for the current video block by comparing one or more reference frames from buffer 213 to the current video block. The motion compensation unit 205 may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from the buffer 213 other than the picture associated with the current video block.

[0066] The motion estimation unit 204 and the motion compensation unit 205 may perform different operations for a current video block, for example, depending on whether the current video block is in an I-slice, a P-slice, or a B-slice. As used herein, an “I-slice” may refer to a portion of a picture composed of macroblocks, all of which are based upon macroblocks within the same picture. Further, as used herein, in some aspects, “P-slices” and “B-slices” may refer to portions of a picture composed of macroblocks that are not dependent on macroblocks in the same picture.

[0067] In some examples, the motion estimation unit 204 may perform uni-directional prediction for the current video block, and the motion estimation unit 204 may search reference pictures of list 0 or list 1 for a reference video block for the current video block. The motion estimation unit 204 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. The motion estimation unit 204 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the

current video block. The motion compensation unit 205 may generate the predicted video block of the current video block based on the reference video block indicated by the motion information of the current video block.

[0068] Alternatively, in other examples, the motion estimation unit 204 may perform
5 bi-directional prediction for the current video block. The motion estimation unit 204 may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. The motion estimation unit 204 may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference
10 video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. The motion estimation unit 204 may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. The motion compensation unit 205 may generate the predicted video block of the current video block based on the reference video blocks
15 indicated by the motion information of the current video block.

[0069] In some examples, the motion estimation unit 204 may output a full set of motion information for decoding processing of a decoder. Alternatively, in some embodiments, the motion estimation unit 204 may signal the motion information of the current video block with reference to the motion information of another video block. For example, the
20 motion estimation unit 204 may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

[0070] In one example, the motion estimation unit 204 may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder 300 that the current video block has the same motion information as the another
25 video block.

[0071] In another example, the motion estimation unit 204 may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video
30 block. The video decoder 300 may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

[0072] As discussed above, video encoder 200 may predictively signal the motion

vector. Two examples of predictive signaling techniques that may be implemented by video encoder 200 include advanced motion vector predication (AMVP) and merge mode signaling.

[0073] The intra prediction unit 206 may perform intra prediction on the current video block. When the intra prediction unit 206 performs intra prediction on the current video block, the intra prediction unit 206 may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

[0074] The residual generation unit 207 may generate residual data for the current video block by subtracting (e.g., indicated by the minus sign) the predicted video block (s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.

[0075] In other examples, there may be no residual data for the current video block for the current video block, for example in a skip mode, and the residual generation unit 207 may not perform the subtracting operation.

[0076] The transform processing unit 208 may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.

[0077] After the transform processing unit 208 generates a transform coefficient video block associated with the current video block, the quantization unit 209 may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

[0078] The inverse quantization unit 210 and the inverse transform unit 211 may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. The reconstruction unit 212 may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the predication unit 202 to produce a reconstructed video block associated with the current video block for storage in the buffer 213.

[0079] After the reconstruction unit 212 reconstructs the video block, loop filtering operation may be performed to reduce video blocking artifacts in the video block.

[0080] The entropy encoding unit 214 may receive data from other functional components of the video encoder 200. When the entropy encoding unit 214 receives the data, the entropy encoding unit 214 may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

[0081] Fig. 3 is a block diagram illustrating an example of a video decoder 300, which may be an example of the video decoder 124 in the system 100 illustrated in Fig. 1, in accordance with some embodiments of the present disclosure.

[0082] The video decoder 300 may be configured to perform any or all of the techniques of this disclosure. In the example of Fig. 3, the video decoder 300 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder 300. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0083] In the example of Fig. 3, the video decoder 300 includes an entropy decoding unit 301, a motion compensation unit 302, an intra prediction unit 303, an inverse quantization unit 304, an inverse transformation unit 305, and a reconstruction unit 306 and a buffer 307. The video decoder 300 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 200.

[0084] The entropy decoding unit 301 may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). The entropy decoding unit 301 may decode the entropy coded video data, and from the entropy decoded video data, the motion compensation unit 302 may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. The motion compensation unit 302 may, for example, determine such information by performing the AMVP and merge mode. AMVP is used, including derivation of several most probable candidates based on data from adjacent PBs and the reference picture. Motion information typically includes the horizontal and vertical motion vector displacement values, one or two reference picture indices, and, in the case of prediction regions in B slices, an identification of which reference picture list is associated with each index. As used herein, in some aspects, a “merge mode” may refer

to deriving the motion information from spatially or temporally neighboring blocks.

[0085] The motion compensation unit 302 may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

[0086] The motion compensation unit 302 may use the interpolation filters as used by the video encoder 200 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. The motion compensation unit 302 may determine the interpolation filters used by the video encoder 200 according to the received syntax information and use the interpolation filters to produce predictive blocks.

[0087] The motion compensation unit 302 may use at least part of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter-encoded block, and other information to decode the encoded video sequence. As used herein, in some aspects, a “slice” may refer to a data structure that can be decoded independently from other slices of the same picture, in terms of entropy coding, signal prediction, and residual signal reconstruction. A slice can either be an entire picture or a region of a picture.

[0088] The intra prediction unit 303 may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. The inverse quantization unit 304 inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and decoded by entropy decoding unit 301.

The inverse transform unit 305 applies an inverse transform.

[0089] The reconstruction unit 306 may obtain the decoded blocks, e.g., by summing the residual blocks with the corresponding prediction blocks generated by the motion compensation unit 302 or intra-prediction unit 303. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in the buffer 307, which provides reference blocks for subsequent motion compensation/intra predication and also produces decoded video for presentation on a display device.

[0090] Some exemplary embodiments of the present disclosure will be described in detailed hereinafter. It should be understood that section headings are used in the present document to facilitate ease of understanding and do not limit the embodiments disclosed in a section to only that section. Furthermore, while certain embodiments are described with reference to Versatile Video Coding or other specific video codecs, the disclosed techniques are applicable to other video coding technologies also. Furthermore, while some embodiments describe video coding steps in detail, it will be understood that corresponding steps decoding that undo the coding will be implemented by a decoder. Furthermore, the term video processing encompasses video coding or compression, video decoding or decompression and video transcoding in which video pixels are represented from one compressed format into another compressed format or at a different compressed bitrate.

1. Initial discussion

[0091] This patent document is related to video coding technologies. Specifically, it is related to the loop filter in image/video coding. The examples may be applied to video coding standard like High-Efficiency Video Coding (HEVC), Versatile Video Coding (VVC), or the standard to be finalized (e.g., third generation audio video standard (AVS3)). The examples may be also applicable to further video coding standards or video codec or be used as post-processing method which is outside of the encoding/decoding process.

2. Further discussion

[0092] Video coding standards have evolved primarily through the development of the well-known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded by VCEG and MPEG jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model (JEM). The Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the VVC standard targeting at 50% bitrate reduction compared to HEVC.

[0093] The Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG is exploring potential neural network video coding technology beyond the capabilities of VVC.

The exploration activities are known as neural network-based video coding (NNVC). The neural network-based (NN-based) coding tools are to enhance or replace conventional modules in the existing VVC design. The implementation of NN-based tools in NNVC 4 are based on Small Ad-hoc Deep Learning (SADL) library.

5 2.1 Definitions of video units

[0094] A picture is divided into one or more tile rows and one or more tile columns. A tile is a sequence of CTUs that covers a rectangular region of a picture. A tile is divided into one or more bricks, each of which comprises a number of CTU rows within the tile.

[0095] A tile that is not partitioned into multiple bricks is also referred to as a brick. However, a brick that is a true subset of a tile is not referred to as a tile. A slice either contains a number of tiles of a picture or a number of bricks of a tile. Two modes of slices are supported, namely the raster-scan slice mode and the rectangular slice mode. In the raster-scan slice mode, a slice contains a sequence of tiles in a tile raster scan of a picture. In the rectangular slice mode, a slice contains a number of bricks of a picture that collectively form a rectangular region of the picture. The bricks within a rectangular slice are in the order of brick raster scan of the slice.

[0096] Fig. 4A shows an example of raster-scan slice partitioning of a picture, where the picture is divided into 12 tiles and 3 raster-scan slices. Specifically, Fig. 4A shows a picture with 18 by 12 luma CTUs that is partitioned into 12 tiles and 3 raster-scan slices.

[0097] Fig. 4B shows an example of a rectangular slice partitioning of a picture, where the picture is divided into 24 tiles (6 tile columns and 4 tile rows) and 9 rectangular slices. Specifically, Fig. 4B shows a picture with 18 by 12 luma CTUs that is partitioned into 24 tiles and 9 rectangular slices.

[0098] Fig. 4C shows an example of a picture partitioned into tiles, bricks, and rectangular slices, where the picture is divided into 4 tiles (2 tile columns and 2 tile rows), 11 bricks (the top-left tile contains 1 brick, the top-right tile contains 5 bricks, the bottom-left tile contains 2 bricks, and the bottom-right tile contain 3 bricks), and 4 rectangular slices. Specifically, Fig. 4C shows a picture that is partitioned into 4 tiles, 11 bricks, and 4 rectangular slices.

2.1.1 CTU/CTB sizes

[0099] In VVC, the CTU size, signaled in SPS by the syntax element `log2_ctu_size_minus2`, could be as small as 4x4.

2.1.2 CTUs in a picture

[0100] Suppose the CTB/LCU size indicated by $M \times N$ (typically M is equal to N , as defined in HEVC/VVC), and for a CTB located at picture (or tile or slice or other kinds of types, picture border is taken as an example) border, $K \times L$ samples are within picture border wherein either
 5 $K < M$ or $L < N$. For those CTBs as depicted in Figs. 4D-4F, the CTB size is still equal to $M \times N$, however, the bottom boundary/right boundary of the CTB is outside the picture.

[0101] Fig. 4D shows an example of CTBs crossing the bottom picture border. Fig. 4E shows an example of CTBs crossing the right picture border. Fig. 4F shows an example of CTBs crossing the right bottom picture border. Accordingly, Figs. 4D-4F show examples of CTBs
 10 crossing picture borders, where in Fig. 4D $K=M$, $L < N$; in Fig. 4E $K < M$, $L=N$; in Fig. 4F $K < M$, $L < N$.

2.2 Coding flow of a typical video codec

[0102] Fig. 5 shows an example of encoder block diagram of VVC, which contains three in-loop filtering blocks: deblocking filter (DF), sample adaptive offset (SAO) and ALF. Unlike
 15 DF, which uses predefined filters, SAO and ALF utilize the original samples of the current picture to reduce the mean square errors between the original samples and the reconstructed samples by adding an offset and by applying a finite impulse response (FIR) filter, respectively, with coded side information signaling the offsets and filter coefficients. ALF is located at the last processing stage of each picture and can be regarded as a tool trying to catch and fix artifacts
 20 created by the previous stages.

2.3 Neural network-based video coding (NNVC)

2.3.1 Neural network-based loop filter set 0

2.3.1.1 Pre-processing and post-processing of chroma

[0103] Fig. 6 illustrates an example of pre-processing and post-processing units. In filter set
 25 0, the filter with a single model is designed to process three components. Since the resolutions of luma and chroma are different, pre-processing and post-processing steps are introduced to up-sample and down-sample chroma components respectively as shown in Fig. 6. In the resampling process, the nearest-neighbor interpolation method is used.

2.3.1.2 Neural network

[0104] Fig. 7 illustrates an example architecture of the CNN in filter set 0. The network structure of the CNN filter is shown in Fig. 7. Along with the reconstructed image (rec_yuv),

additional side information is also fed into the network, such as the prediction image (pred_yuv), slice QP, base QP and slice type. In the ResBlock, the number of channels firstly goes up before the activation layer, and then goes down after the activation layer. Specifically, K and M are set to 64 and 160 respectively, and the number of Resblock is set to 32.

5 2.3.1.3. Combination with conventional filters

[0105] Fig. 8 illustrates an example implementation of the CNN in filter set 0. As shown in Figure.3, the reconstructed samples before DBK are fed into the CNN based filter (CNNLF), then final filtered samples are generated by blending the result of CNNLF and SAO. This blending process can be briefly formulated as:

$$R_{Blend} = w \times R_{NN} + (1 - w) \times R_{SAO}$$

[0106] There are four candidates, 1, 0.75, 0.5 and an adaptive weight, for the blending weight. With regard to the adaptive weight, its derivation is based on least square method. If the adaptive weight is selected, the blending weight is signaled for each color component in the slice header.

15 2.3.1.4. Mode selection

[0107] The CNN filter can be turned on/off at the CTU level and slice level. For each enabling type, there are four blending ways. Therefore, there are nine modes to be evaluated by RDO at encoder. The final selected mode would be signaled in the slice header.

Table 1. Parameter selection of filter set 0

Mode	On/off type	Blending weight (w)
0	Disable at slice level	None
1	Enable at slice level	Adaptive weight
2		1
3		0.75
4		0.5
5	Enable at CTU level	Adaptive weight
6		1
7		0.75

8		0.5
---	--	-----

2.3.1.5 Base QP adjustment

[0108] Base QP is fed into the CNN filter as shown in Figure. 8. To improve adaptation, an offset can be added to the base QP (the adjusted base QP is used as the input to the NN filter) at slice level. The offset candidates are $\{-5, 5\}$. For example given the offset -5, the actual input base QP to the filter becomes (BaseQP - 5) for the current slice.

Encoder approach

[0109] Fig. 9 illustrates an example encoder optimization. An example encoder only filters one out of every four CTUs during the process of selecting the best base QP offset to save encoding time. As shown in Fig. 9, only shaded CTUs are considered for calculating distortions of using different BaseQP candidates $\{\text{BaseQP}, \text{BaseQP}-5, \text{BaseQP}+5\}$. After the candidate with the smallest cost is selected, the encoder filters the rest of CTUs (non-shaded ones in Fig. 9) by applying the best offset to the base QP.

2.3.1.6. Encoder-only Optimization

[0110] To more accurately estimate the rate-distortion (RD) cost with integrated NN-based in-loop filters, an encoder-only NN filter is involved in the partitioning decision process. In the partitioning mode decision, the distortion between NN filtered samples and original samples is calculated, and then the optimal partitioning mode is selected based on calculated distortion to make the partitioning decision more accurate. To reduce complexity, only few ResBlocks (see Section 3.1.2.) are used in the network structure. The NN filter in the RDO process is implemented with SADL using int16 precision. This encoder-only NN tool is disabled by default.

2.3.1.7 Inference details

[0111] SADL is used for performing the inference of the CNN filters. Both floating point-based and fixed point-based implementations are supported. In the fixed-point implementation, both weights and feature maps are represented with int16 precision using a static quantization method. The network information in the inference stage is provided in Table 2.

Table 2. Network Information of filter set 0 in Inference Stage

<u>Network Information in Inference Stage</u>	
Mandatory	HW environment:

	GPU Type	N/A
	Framework:	SADL
	Number of GPUs per Task	0
	Number of Parameters (Each Model)	1.9M
	Total Parameter Number	1.9M, one model in total
	Parameter Precision (Bits)	float: 32 int: 16
	Memory Parameter (MB)	float: 7.6MB, one model in total int: 3.8MB, one model in total
	Multiply Accumulate (kMAC/pixel)	485 (assuming frame-level input) 615 (assuming block-level input)
Optional		
	Total Conv. Layers	101
	Total FC Layers	0
	Total Memory (MB)	
	Batch size:	1
	Patch size	144×144
	Changes to network configuration or weights required to generate rate points	
	Peak Memory Usage	
	Other information:	

2.3.2. Neural network-based loop filter set 1

2.3.2.1. Neural network for luma component

[0112] Figs. 10A-10C illustrate an example architecture of the CNN in filter set 1. Fig. 10A illustrates an example head of luma network. The inputs are combined to form the input y to the next part of the network. Fig. 10B illustrates an example subnetwork. The k-th residual block (k=0..7). The output y of the head is fed into a first residual block with input z0=y. The output z1 is then fed into another such residual block. Fig. 10C illustrates another example subnetwork. The output of the last residual block is fed into this last part of the network.

[0113] There are two regular networks in filter set 1, one for luma and one for chroma. The inputs of the luma network comprise the reconstructed luma samples (rec), the prediction luma samples (pred), boundary strengths (bs), QP, and the block type (IPB). The numbers of feature maps and residual blocks are set as 96 and 8 respectively. The structure of the luma network is depicted in Fig. 10A-10C.

2.3.2.23 Neural network for chroma component

[0114] Luma information is taken as additional input for the in-loop filtering of chroma. Considering the resolution of luma is higher than chroma in YUV 4:2:0 format, features are first extracted separately from luma and chroma. Then luma features are down-sampled and concatenated with chroma features. The inputs of the chroma network include reconstructed luma samples (recY), reconstructed chroma samples (recUV), predicted chroma samples (predUV), boundary strength (bsUV), and QP. Regarding network backbone, chroma components use the same one as luma.

2.3.2.3. Temporal filter

[0115] Fig. 11 illustrates an example temporal in-loop filter. Only the head part is illustrated. Other parts remain the same as in Fig. 10B-C. {Col 0, Col 1} refers to collocated samples from the first picture in both reference picture lists. Filter set 1 contains an additional in-loop filter, namely temporal filter, which takes collocated blocks from the first picture in both reference picture lists to improve performance. The two collocated blocks are directly concatenated and fed into the network as shown in Fig. 11. When enabling temporal filtering feature, the temporal filter is applied to the luma component of pictures in three highest temporal layers, while the regular luma and chroma filters are used for other cases. By default, this temporal filtering feature is disabled.

2.3.2.4. Adaptive inference granularity

[0116] The granularity of the filter determination and the parameter selection is dependent on resolution and QP. Given a higher resolution and a larger QP, the determination and selection are performed in a larger region.

2.3.2.5. Parameter selection

[0117] Each slice or block, a determination can be made whether to apply the CNN-based filter or not. When the CNN-based filter is determined to be applied to a slice/block, which conditional parameter from a candidate list including three candidates derived from QP could be further decided. The sequence level QP is denoted as q , and the candidate list includes conditional parameters {Param_1, Param_2, Param_3}. For low temporal layers, Param_1 = q ,

Param_2 = q-5, Param_3 = q-10. For high temporal layers, Param_1 = q, Param_2 = q-5, Param_3 = q+5. In other words, the third candidate is different across different temporal layers.

[0118] Fig. 12A illustrates an example parameter selection at an encoder side. Fig. 12B illustrates an example parameter selection at a decoder side. The selection process is based on the rate-distortion cost at the encoder side. Indication of on/off control as well as the conditional parameter index, if needed, are signalled in the bitstream. Figs. 12A-12B show the diagram of parameter selection at encoder and decoder sides. All blocks in the current frame need to be processed with three conditional parameters first. Then five costs, i.e. Cost_0, ..., Cost_5, are calculated and compared against each other to achieve optimum rate-distortion performance. In Cost_0, CNN-based filter is prohibited for all blocks. In Cost_i, {i = 1, 2, 3}, the parameter Param_i is used for all blocks. In Cost_4, different blocks may prefer different parameters, and the information regarding whether to use CNN-based filter or which parameter to be used is signaled for each block. At decoder side, whether to use CNN-based filter or which parameter to be used for a block is based on the Param_Id parsed from the bit-stream as shown in Fig. 12B.

[0119] For all-intra configuration, parameter selection is disabled while filter on/off control is still preserved. A shared conditional parameter is used for the two chroma components to ease the burden in worst case at decoder side. In addition, the max number of conditional parameter candidates could be specified at encoder side.

2.3.2.6. Residue scaling

[0120] When a NN filter is being applied to reconstructed pictures, a scaling factor is derived and signaled for each color component in the slice header. The derivation is based on least square method. The difference between the input samples and the NN filtered samples (residues) are scaled by the scaling factors before being added to input samples.

2.3.2.7. Combination with deblocking filter

[0121] To enable a combination with deblocking, the input samples used in the residual scaling is the output of deblocking filtering. The residual scaling process is shown below, where R_{NN} and R_{DB} refer to the outputs of NN filtering and deblocking filtering respectively.

$$R_{Refine} = (R_{NN} - R_{DB}) \times w + R_{DB} = w \times R_{NN} + (1 - w) \times R_{DB}$$

2.3.2.8. Encoder-only optimization

[0122] Different from NNVC-2.0, EncDbOpt is also enabled for AI configuration. For a better estimation of rate-distortion (RD) cost in the case the NN filter is used, an example encoder introduces NN-based filtering into the rate-distortion optimization (RDO) process of

partitioning mode selection. Specifically, a refined distortion is calculated by comparing the NN filtered samples and the original samples. The partitioning mode with the smallest rate-refined distortion cost is selected as the optimal one. To reduce complexity, several fast algorithms are applied. First, NN model is simplified by using a smaller number of residual blocks. Second, parameter selection is not allowed for the NN filtering in the RDO process. Third, the disclosed technique is only applied to the coding units with height and width no larger than 64. The NN filter used in the RDO process is also implemented with SADL using fixed point-based calculation. This NN-based encoder-only method is disabled by default.

2.3.2.9. Inference details

[0123] SADL is used for performing the inference of the CNN filters. Both floating point-based and fixed point-based implementations are supported. In the fixed-point implementation, both weights and feature maps are represented with int16 precision using a static quantization method. The network information in the inference stage is provided in Table 3.

Table 3. Network Information of filter set 1 in Inference Stage

<u>Network Information in Inference Stage</u>		
Mandatory	HW environment:	
	GPU Type	N/A
	Framework:	SADL
	Number of GPUs per Task	0
	Total Parameter Number	1.55M/model, 2 models in total for all tests
	Parameter Precision (Bits)	float: 32 int: 16
	Memory Parameter (MB)	float: 6.2MB/model, 2 models int: 3.1MB/model, 2 models
	Multiply Accumulate (kMAC/pixel)	532 (assuming frame-level input) 673 (assuming block-level input)
Optional		
	Total Conv. Layers	25
	Total FC Layers	0
	Total Memory (MB)	

	Batch size:	1
	Patch size	144×144, 272×272
	Changes to network configuration or weights required to generate rate points	
	Peak Memory Usage	
	Other information:	

2.3.3. Neural network-based intra prediction

2.3.3.1. Neural network inference

[0124] Fig. 13 illustrates prediction of a current $w \times h$ block Y from the context X of reference samples around Y via the neural network-based intra prediction mode. Here, $w = 8$ and $h = 4$. The neural network-based intra prediction mode contains 7 neural networks, each predicting blocks of a different size in $\{4 \times 4, 8 \times 4, 16 \times 4, 32 \times 4, 8 \times 8, 16 \times 8, 16 \times 16\}$. The neural network predicting blocks of size $w \times h$ is denoted $f_{h,w}(\cdot, \theta_{h,w})$ where $\theta_{h,w}$ gathers its parameters. For a given $w \times h$ block Y , $f_{h,w}(\cdot, \theta_{h,w})$ takes a preprocessed version $\tilde{X}$ of the context X made of n_a rows of $n_l + 2w + e_w$ reference samples located above this block and n_l columns of $2h + e_h$ reference samples on its left side to provide $\tilde{Y}$. The application of a postprocessing to $\tilde{Y}$ yields a prediction $\hat{Y}$ of Y , see Fig. 13. Besides, $f_{h,w}(\cdot, \theta_{h,w})$ returns two indices grpIdx_1 and grpIdx_2 . grpIdx_i denotes the index characterizing the LFNST kernel index and whether the primary transform coefficients resulting from the application of the DCT-2 horizontally and the DCT-2 vertically to the residue of the neural network prediction are transposed when $\text{lnstIdx} = i$, $i \in \{1, 2\}$, see Fig. 13. Furthermore, $f_{h,w}(\cdot, \theta_{h,w})$ gives the index $\text{repIdx} \in \llbracket 0, 66 \rrbracket$ of the VVC intra prediction mode (PLANAR or DC or directional intra prediction mode) whose prediction of Y from the reference samples surrounding Y best represents $\hat{Y}$, see Fig. 13.

20 If $\min(h, w) \leq 8$ && $hw < 256$:

$$n_a = n_l = \min(h, w)$$

otherwise:

if $h > 8$:

$$n_a = h/2$$

25 otherwise:

$$n_a = h$$

if $w > 8$:

$$n_l = w/2$$

otherwise:

$$n_l = w$$

5 If $h \leq 8$, $e_h = 4$. Otherwise, $e_h = 0$.

If $w \leq 8$, $e_w = 4$. Otherwise, $e_w = 0$.

2.3.3.2. Preprocessing and postprocessing

2.3.3.2.1. Preprocessing of the context of the current block

[0125] The preprocessing shown in Fig. 13 comprises the four following steps.

- 10 • The mean μ of the available reference samples $\bar{X}$ in X , see Fig. 3, is subtracted from $\bar{X}$.
- If the neural network predicting the current block is in floats, the reference samples in the context X are multiplied by $\rho = 1/(2^{b-8})$, b being the internal bitdepth, i.e. 10 in VVC. Otherwise, the reference samples in the context X are multiplied by $\rho = 2^{Q_{in}-b+8}$, Q_{in} denoting the input quantizer.
- 15 • All the unavailable reference samples X_u in X , see Fig. 3, are set to 0.
- The context resulting from the previous step is flattened, yielding $\tilde{X}$, a vector of size $n_a(n_l + 2w + e_w) + (2h + e_h)n_l$.

[0126] Fig. 14 illustrates decomposition of a context X of reference samples surrounding the current $w \times h$ block Y into the available reference samples $\bar{X}$ and the unavailable reference samples X_u . Here, $w = 8$ and $h = 4$. In the illustrated case, the number of unavailable reference samples reaches its maximum value.

2.3.3.2.2. Postprocessing of the neural network prediction

[0127] The postprocessing depicted in Fig. 13 comprises reshaping the vector $\tilde{Y}$ of size hw into a rectangle of height h and width w , dividing the result of the reshape by ρ , adding the mean μ of the available reference samples in the context of the current block, and clipping to $[0, 2^b - 1]$. Therefore, the postprocessing can be summarized as:

$$\hat{Y} = \min \left(\max \left(\frac{\text{reshape}(\tilde{Y})}{\rho} + \mu, 0 \right), 2^b - 1 \right).$$

2.3.3.3. Adaptation of the derivation of the list of MPMs

[0128] When creating the MPM list of a given luma CB, if the “left” luma CB is predicted via the neural network-based intra prediction mode, the neural network-based mode index can be replaced by the replIdx returned during the prediction of the “left” luma CB and become a candidate index to be put into the MPM list. Similarly, if “above” luma CB is predicted via the

neural network-based intra prediction mode, the neural network-based mode index can be replaced by the `repIdx` returned during the prediction of the “above” luma CB and become a candidate index to be inserted into the MPM list.

2.3.3.4. Signaling of the neural network-based intra prediction mode

5 2.3.3.4.1. Signaling of the neural network-based intra prediction mode in luma

[0129] Fig. 15 illustrates intra prediction mode signaling for the current $w \times h$ luma CB framed in the dashed line. The coordinates of the pixel at the top-left of this CB are (y, x) . The bin value of a `nnFlag` value appears in bold black. Here, $h = 8$, $w = 4$, $x = 8$, and $y = 0$.

10 [0130] For the current $w \times h$ luma CB whose top-left pixel is at position (y, x) in the current luma channel, the intra prediction mode signaling in luma is split into two cases.

- If $(h, w) \in T$, `nnFlag` appears in the intra prediction mode signaling in luma. `nnFlag = 1` means that the neural network-based intra prediction mode is selected to predict the current luma CB and END. `nnFlag = 0` means that the neural network-based intra prediction mode is not selected to predict the current luma CB, then the regular intra prediction mode signaling in luma, denoted S_c , applies, see Fig. 15.

- Otherwise, the regular intra prediction mode signaling in luma S_c applies.

[0131] Note that, in the case “ $(h, w) \in T \ \&\& \ nnFlag = 1$ ”, if the context of the current luma CB goes out of the bounds of the current luma channel, i.e. $x < n_l \ || \ y < n_a$, the neural network-based intra prediction is replaced by PLANAR.

20 $T = \{(4,4), (4,8), (8,4), (4,16), (16,4), (4,32), (32,4), (8,8), (8,16), (16,8), (8,32), (32,8), (16,16), (16,32), (32,16), (32,32), (64,64)\}$.

2.3.3.4.2. Signaling of the neural network-based intra prediction mode in chroma

[0132] For the current $w \times h$ chroma CB whose top-left pixel is at position (y, x) in the current chroma channel, the intra prediction mode signaling in chroma is split into two cases.

- 25 • If the luma CB collocated with this chroma CB is predicted by the neural network-based intra prediction mode:
 - If $(h, w) \in T$, the DM becomes the neural network-based intra prediction mode.
 - Otherwise, the DM is set to PLANAR.
- Otherwise:
 - 30 ○ If $(h, w) \in T$, `nnFlagChroma` appears in the intra prediction mode signaling in chroma. `nnFlagChroma` is placed before the DM flag in the decision tree of the intra prediction mode signaling in chroma. `nnFlagChroma = 1` means that the neural network-based intra prediction mode is selected to predict the current pair

of chroma CBs and END. $nnFlagChroma = 0$ means that the neural network-based intra prediction mode is not selected to predict the current pair of chroma CBs, then the regular intra prediction mode signaling in chroma resumes from the DM flag.

- 5 ○ Otherwise, the regular intra prediction mode signaling in chroma applies.

[0133] Note that, in the case where “ $(h, w) \in T$ and the DM becomes the neural network – based intra prediction mode” and the case where “ $(h, w) \in T \ \&\& \ nnFlagChroma = 1$ ”, if the context of the current chroma CB goes out of the bounds of the current chroma channel, i.e. $x < n_l \ || \ y < n_a$, the neural network-based intra prediction is replaced by PLANAR.

2.3.3.5. Transformation of the context and the neural network prediction

[0134] For a given $w \times h$ block, if $(h, w) \in T$, it is possible that the neural network-based intra prediction mode must predict this block but the neural network-based intra prediction mode does not contain $f_{h,w}(\cdot, \theta_{h,w})$. In this case, the context of the current block can be down-sampled vertically by a factor δ and/or down-sampled horizontally by a factor γ and/or transposed before the step called “preprocessing” in Fig. 13. Then, the prediction of the current block can be transposed and/or up-sampled vertically by the factor δ and/or up-sampled horizontally by the factor γ after the step called “postprocessing” in Fig. 13. The transposition of the context of the current block and the prediction, δ , and γ are chosen so that a neural network belonging to the neural network-based intra prediction mode is used for prediction, see Table 4.

height and width of the block to be predicted (h, w)	γ	δ	transposition	neural network used for prediction
(4, 4)	1	1	no	$f_{4,4}(\cdot, \theta_{4,4})$
(4, 8)	1	1	no	$f_{4,8}(\cdot, \theta_{4,8})$
(8, 4)	1	1	yes	$f_{4,8}(\cdot, \theta_{4,8})$
(4, 16)	1	1	no	$f_{4,16}(\cdot, \theta_{4,16})$
(16, 4)	1	1	yes	$f_{4,16}(\cdot, \theta_{4,16})$
(4, 32)	1	1	no	$f_{4,32}(\cdot, \theta_{4,32})$
(32, 4)	1	1	yes	$f_{4,32}(\cdot, \theta_{4,32})$

(8, 8)	1	1	no	$f_{8,8}(\cdot, \theta_{8,8})$
(8, 16)	1	1	no	$f_{8,16}(\cdot, \theta_{8,16})$
(16, 8)	1	1	yes	$f_{8,16}(\cdot, \theta_{8,16})$
(8, 32)	2	1	no	$f_{8,16}(\cdot, \theta_{8,16})$
(32, 8)	1	2	yes	$f_{8,16}(\cdot, \theta_{8,16})$
(16, 16)	1	1	no	$f_{16,16}(\cdot, \theta_{16,16})$
(16, 32)	2	1	no	$f_{16,16}(\cdot, \theta_{16,16})$
(32, 16)	1	2	no	$f_{16,16}(\cdot, \theta_{16,16})$
(32, 32)	2	2	no	$f_{16,16}(\cdot, \theta_{16,16})$
(64, 64)	4	4	no	$f_{16,16}(\cdot, \theta_{16,16})$

Table 4: decision of transposing the context of the current $w \times h$ block to be predicted and the prediction of this block, the value of γ , and the value of δ , and the neural network belonging to the neural network-based intra prediction mode used for prediction for each $(h, w) \in T$.

5 2.3.4. Small ad-hoc deep learning (SADL) library

[0135] SADL (Small Ad-hoc Deep-Learning Library) is a header only small library for inference of neural networks. SADL provides both floating-point-based and integer-based inference capabilities. The inference of neural networks in NNVC is based on the SADL.

[0136] The table below summarizes the framework characteristics.

10

Table 5. Characteristics of SADL

Language	Pure C++, header only.
Footprint	~6000 LOC, library ~300kB, no dependency
Optimization	Some SIMD at hot spots, e.g. convolution (conv2D) and automatic sparse matrix-vector multiplication
Compatibility	Onnx to SADL converter
Layer Supports	constants, add, maxPool, matMul (dense and sparse), reshape, ReLU, conv2D (strided, grouped, separated), mul, concat, max, leakyReLU, shape, expand, PReLU, flatten, transpose, Cond2DTranspose, Slicing
Type support	float, int32, int16, int8
Quantization	Support adaptive quantizer per layer
License	BSD 3-Clause

[0137] NNV repository uses SADL as a submodule.

3. Technical problems solved by disclosed technical solutions

[0138] According to the disclosure, an example NN-based loop filtering has the following problems:

[0139] First, the same convolutions are used for each input of NN-based loop filter. However, different inputs may have different importance. Therefore, it is reasonable to assign different convolutions with different kernel size and channel numbers for each input.

[0140] Second, the convolution with kernel size $K \times K$ is widely used in NN-based loop filter. However, it can be decomposed into combinations of several convolutions with smaller kernel size to reduce the complexity.

[0141] Third, side information generated during compression may be used as extra input to improve the performance of NN-based loop filter. For example, prediction picture, slice type, boundary strength, base QP, slice QP, and IPB information could be used as side information.

[0142] Fourth, multi-scale structure for neural network may be beneficial for improving the performance of NN-based loop filter.

[0143] Fifth, the NN-based network is used for designing loop filter. However, non-adjacent information is not considered. The Transformer based network could capture global information.

[0144] Sixth, the NN-based filter is proposed to enhance reconstruction. However, the example filter may exceed the NN-based filter for some video content. So it is reasonable to combine the standardized filter and NN filter.

[0145] Seventh, the parameters of NN-based filter are fixed in the training process. However, the parameter such as QP, inference size, block extension size should adapt to the various video content in the inference process.

[0146] Eighth, the NN-based loop filter may be used for luma component, chroma component, or both luma and chroma components.

[0147] Ninth, the NN-based loop filter may be used for intra slice, inter slice, or both intra and inter slices.

4. A listing of solutions and embodiments

[0148] The detailed embodiments below should be considered as examples to explain general concepts. These examples should not be interpreted in a narrow way. Furthermore, these embodiments can be combined in any manner.

[0149] One or more neural network (NN) filter models are trained as part of an in-loop filtering technology or filtering technology used in a post-processing stage for reducing the distortion incurred during compression. Samples with different characteristics are processed by different NN filter models. This disclosure elaborates how to design a unified NN filter model by feeding at least one indicator which may be related to the quality level (e.g. QP or Constant rate factor (CRF) value or bitrates)/slice type/coding modes/coded information as the input of NN filter.

[0150] It should be noted that the concept of unifying NN model by feeding the indicator as the input of NN process could be also extended to other NN-based coding tools, such as NN-based intra prediction, NN-based cross component prediction, NN-based inter prediction, NN-based super-resolution, NN-based motion compensation, NN-based transform design. In the examples below, NN-based filtering technology is used as an example.

[0151] It should also be noted that the concept of feeding the coded information as the input of NN process could be also extended to non-NN-based coding tools, such as non-NN-based intra prediction, non-NN-based cross component prediction, non-NN-based inter prediction, non-NN-based super-resolution, non-NN-based motion compensation, non-NN-based transform design. For example, a non-NN based coding tool may classify the to-be-filtered samples into different categories using the coded information.

[0152] In the disclosure, a NN filter can be any kind of NN filter, such as a convolutional neural network (CNN) filter, fully connected neural network filter, transformer-based filter, recurrent neural network-based filter.

[0153] In the following discussion, a video unit may be a sequence, a picture, a slice, a tile, a brick, a subpicture, a CTU/CTB, a CTU/CTB row, one or multiple CUs/CBs, one or multiple CTUs/CTBs, one or multiple VPDU (Virtual Pipeline Data Unit), a sub-region within a picture/slice/tile/brick. A father video unit represents a unit larger than the video unit. Typically, a father unit will contain several video units. E.g., when the video unit is CTU, the father unit could be slice, CTU row, multiple CTUs, etc.

To simplify the NN filter by adaptive channel number

1. To solve problem 1, different convolution types are assigned to different inputs.

a. In one example, the convolution shares the same convolution kernel size and different channel numbers may be assigned for each input.

i. In one example, the channel numbers for each input are denoted as $C_1, C_2, \dots, C_n$ where n is an integer which denotes the number of inputs.

- 1) In one example, furthermore, one constraint is added that for any indexes i and j where $1 \leq i, j \leq n$ that $C_i \neq C_j$.
 - 2) In one example, furthermore, one constraint is added that there is at least one pair of indexes i and j where $1 \leq i, j \leq n$ that $C_i \neq C_j$.
- 5 b. In one example, the convolution shares the same convolution channel numbers and different convolution kernel size are assigned for each input.
- i. In one example, 1×1 kernel size may be used for the convolution of partial input and $K \times K$ kernel size is used for the convolution of the rest input. The K denotes an integer value greater than 1.
- 10 c. In one example, different convolution channel numbers and different convolution kernel size are assigned for each input.

To simplify the NN filter by decomposition

2. To solve problem 2, the $C_1 \times C_2 \times K \times K$ convolution may be decomposed into a combination of several convolutions with smaller kernel size. The K denotes an integer value greater than 1, C_1 and C_2 denote the input channel number and output channel number of the convolution, respectively.
- 15 a. In one example, the input and output channel number of the convolution may be not changed when it is decomposed.
- i. In one example, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_2 \times 1 \times K$ convolution followed by $C_1 \times C_2 \times K \times 1$ convolution.
 - 20 ii. In one example, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_2 \times 1 \times K$ convolution followed by $C_1 \times C_2 \times K \times 1$ convolution and any activation layer may be placed after each convolution.
 - 25 iii. In one example, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_2 \times K \times 1$ convolution followed by $C_1 \times C_2 \times 1 \times K$ convolution.
 - iv. In one example, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_2 \times K \times 1$ convolution followed by $C_1 \times C_2 \times 1 \times K$ convolution and any activation layer may be placed after each convolution.
 - 30 b. In one example, the input and output channel number of the convolution may be changed when it is decomposed.

- i. In one example, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_3 \times 1 \times K$ convolution followed by $C_3 \times C_2 \times K \times 1$ convolution, where C_3 is a postive integer different with C_2 .
- 5 ii. In one example, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_3 \times 1 \times K$ convolution followed by $C_3 \times C_2 \times K \times 1$ convolution and any activation layer may be placed after each convolution, where C_3 is a postive integer different with C_2 .
- 10 iii. In one example, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_3 \times K \times 1$ convolution followed by $C_3 \times C_2 \times 1 \times K$ convolution, where C_3 is a postive integer different with C_2 .
- iv. In one example, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_3 \times K \times 1$ convolution followed by $C_3 \times C_2 \times 1 \times K$ convolution and any activation layer may be placed after each convolution, where C_3 is a postive integer different with C_2 .
- 15 c. In one example, partial $K \times K$ convolutions are decomposed.
- d. In one example, all the $K \times K$ convolutions are decomposed.

To design side information of NN filter

- 3. To solve problem 3, which side information to be used as extra input of NN-based loop filter and how they are processed is specified.
- 20 a. In one example, the side information may be used as extra input of NN-based loop filter.
 - i. In one example, the slice QP may be used as extra input of NN-based loop filter.
 - 1) In one example, furthermore, the slice QP is normalized by $\text{sliceQP} \div \text{MAX_QP}$ where the value of MAX_QP may be 63.
 - 25 2) In one example, furthermore, the slice QP is first tiled or spanned into a 2-dimensional arrays with the same size as the video unit to be filtered.
 - ii. In one example, the base QP may be used as extra input of NN-based loop filter.
 - 30 1) In one example, furthermore, the base QP is normalized by $\text{baseQP} \div \text{MAX_QP}$ where the value of MAX_QP may be 63.

- 2) In one example, furthermore, the base QP is first tiled or spanned into a 2-dimensional arrays with the same size as the video unit to be filtered.
- iii. In one example, the prediction picture may be used as extra input of NN-based loop filter.
- iv. In one example, the slice type may be used as extra input of NN-based loop filter.
- 1) In one example, furthermore, the slice type may be a binary value which indicates whether the picture to be filtered is intra slice.
- 2) In one example, the slice type indicator is first tiled or spanned into a 2-dimensional arrays with the same size as the video unit to be filtered.
- v. In one example, the IPB information of the video unit to be filtered may be used as extra input of NN-based loop filter.
- 1) In one example, furthermore, the IPB information may be derived from the slice type.
- a. In one example, the value of IPB information is derived as follows, it is equal to a if slice type is I slice, or it is equal to b if slice type is B slice, or it is equal to c if slice type is P slice, where a, b, and c are constants.
- i. In one example, $b = c$.
- ii. In one example, $b = -a$, $c = -a$.
- iii. In one example, $a = 1$, $b = -1$, $c = -1$.
- iv. In one example, $a = 1$, $b = 0.5$, $c = 0.5$.
- 2) In one example, the IPB information is first tiled or spanned into a 2-dimensional arrays with the same size as the video unit to be filtered.
- vi. In one example, the boundary strength of the video unit to be filtered may be used as extra input of NN-based loop filter.
- vii. In one example, any combination of above side information may be used as extra input of NN-based loop filter.
- b. In one example, convolutions for each input side information are performed separately and then all the convolution results are concatenated with the output of convolution of reconstruction picture.

- c. In one example, the reconstruction picture and side information are concatenated and followed by a convolution.
- 4. To solve problem 4, multi-scale neural network structure may be used in the NN-based loop filter.
- 5 a. In one example, neural network with two branches may be used and one possible solution is presented as follows. Fig. 16 illustrates an example multi-scale neural network with two branches for use in a NN-based loop filter. In Fig. 16, K_i denotes the kernel size where $1 \leq i \leq 4$, C_j is related to the channel numbers, where $1 \leq j \leq 8$.
- 10 i. In one example, further more, the convolution with kernel size $K \times K$ may be decomposed into several convolutions with smaller kernel size, which is a combination with bullets 2.
- ii. In one example, the activation layers may be used.
 - 1) In one example, furthermore, the activation layers may be used after the activation layers may be used after all the convolution layers.
 - 15 2) In one example, furthermore, the activation layers may be used after the activation layers may be used after partial the convolution layers.
 - 3) In one example, the activation layers might be PRelu layer.
 - 4) In one example, the activation layers might be Relu layer.
- 20 iii. In one example, the residual connection may be used.
- b. In one example, neural network with multiple branches may be used. In each branch, different convolution with different kernel size, channel number, or stride might be used.
- 25 i. In one example, further more, the convolution with kernel size $K \times K$ may be decomposed into several convolutions with smaller kernel size, which is a combination with bullets 2.
- ii. In one example, the activation layers may be used.
 - 1) In one example, furthermore, the activation layers may be used after the activation layers may be used after all the convolution layers.
 - 30 2) In one example, furthermore, the activation layers may be used after the activation layers may be used after partial the convolution layers.
 - 3) In one example, the activation layers might be PRelu layer.
 - 4) In one example, the activation layers might be Relu layer.
- iii. In one example, the residual connection may be used.

c. In one example, this multi-scale neural network structure may be used as basic block in the whole NN-based loop filter.

i. In one example, this basic block may be used N times where N is an integer greater than 0.

5 ii. In one example, this basic block may be used together with other kinds of blocks.

1) In one example, this basic block may be used together with other convolution layers and activation layers.

10 2) In one example, this basic block may be used together with residual blocks.

To design Transformer based NN filter

5. To solve problem 5, it is proposed that Transformer-based structure may be involved in the design of the NN filter.

15 a. In one example, the head/backbone/tail may be designed by using Transformer network.

6. To solve problem 5, it is proposed that Transformer-based structure may be combined with CNN in the design of the NN filter.

a. In one example, CNN module may be followed by Transformer module in the network of NN filter.

20 7. To solve problem 5, it is proposed that transformer-based structure and CNN-based structure may be alternative in the design of the NN filter.

a. In one example, the NN filters may be selected when there are two NN-based filters which contain one CNN-based filter and one transformer-based filter.

To combine NN filter with DBF/SAO

25 8. To solve problem 6, it is proposed that the NN filter may be combined/fused/blended with traditional filters.

a. In one example, traditional filters may be Deblocking filter (DBF).

b. In one example, traditional filters may be Sample Adaptive Offset (SAO).

c. In one example, traditional filters may be the combination of DBF and SAO.

30 d. In one example, the NN filter and DBF may be fused/blended.

i. In one example, SAO may be after the fusing/blending of NN filter and DBF filter.

1) In one example, SAO may be enabled/disabled by the syntax element in the SPS/PPS, etc.

ii. In one example, ALF may be used after SAO.

1) In one example, ALF may be enabled/disabled by the syntax element in the SPS/PPS, etc.

5

e. In one example, the NN filter and the combination of DBF and SAO may be fused/blended.

i. In one example, Adaptive Loop Filter (ALF) may be used after the fusing/blending of NN filter and the combination of DBF and SAO.

10

1) In one example, ALF may be enabled/disabled by the syntax element in the SPS/PPS, etc.

To design scaling factor

9. To solve problem 6, it is proposed that the reconstruction samples of traditional filter and NN filter are fused/blended by a scaling factor.

15

- a. In one example, reconstruction samples may be fused/blended in slice/block level.
- b. In one example, the scaling factor may be adaptive which can be determined by video content.
- c. In one example, the scaling factor may be pre-defined.
- d. In one example, the scaling factor may be separated for different components.

20

- i. In one example, the scaling factor may be separated for luma and chroma components.
- ii. In one example, the scaling factor may be separated for U and V components.

To design QP adjustment, parameter number, block QP adaptation

10. To solve problem 7, it is proposed that there is a candidate list which contains multiple input parameters.

25

- a. In one example, the number of candidates in the list may be configurable in sequence/slice/block level.
- b. In one example, the candidate list may be constructed in sequence/slice/block level.
- c. In one example, the candidate list may be adaptive for sequence/slice/block level.
- d. In one example, the input parameter is a variable dependent on base QP which denoted as q .

30

- i. In one example, q is QP in sequence/slice/block level.
- e. In one example, the candidate list may include the base QP and adjusted QP.

- i. In one example, adjusted QP may be equal to the base QP q by adding an offset.
- ii. In one example, offset may be equal to -5, 10, 5.
- iii. In one example, offset may be dependent on the TID level.

5 To design adaptive inference granularity/ configurable inference size

11. To solve problem 7, it is proposed that the inference granularity/size of NN filter may be adaptive for the sequence/slice/block level.

- a. In one example, the granularity/size may be dependent on the slice type.
- b. In one example, the granularity/size may be configurable which dependent on the syntax element in the sequence/slice/block level.

To design block extension size

12. To solve problem 7, it is proposed that the block extension/padding size may be adaptive for the sequence/slice/block level.

- a. In one example, the block extension/padding size may be dependent on the block modes and/or slice type.
- b. In one example, the block extension/padding size may be configurable which dependent on the syntax element in the sequence/slice/block level.

To the usage of unified NN filter

13. To solve problem 8, the output of NN-based filter may be luma components, chroma components, or both luma and chroma components.

- a. In one example, separate NN-based filters may be used for generating outputs of luma components and outputs of chroma components, respectively.
 - i. In one example, furthermore, a total number of M_1 models may be used for generating output of luma components, and a total number of M_2 models may be used for generating outputs of chroma components, where both M_1 and M_2 are integers greater than 0.

1) In one example, furthermore, the Cb and Cr components may share the same chroma model of NN-based filter.

- b. In one example, a single NN-based filters may be used for generating outputs of luma components and outputs of chroma components.

- 5
- i. In one example, furthermore, a total number of M_1 models may be used, and for each model, the outputs of luma components and chroma components may be generated, where M_1 is an integer greater than 0.
 - ii. In one example, furthermore, the output luma and chroma components generated by the single NN-based filter may be used together.
 - iii. In one example, furthermore, the output luma and chroma components generated by the single NN-based filter may be used separately.
 - 1) In one example, furthermore, the output luma component generated by the single NN-based filter may be used and the output luma component generated by the single NN-based filter may be NOT used.
 - 2) In one example, furthermore, the output luma component generated by the single NN-based filter may be NOT used and the output luma component generated by the single NN-based filter may be used.
- 10
- 15 14. To solve problem 8, the single NN-based filter may generate the outputs of both luma and chroma components.
- a. In one example, one or more syntax elements may be signaled to control the usage of NN-based filter in the slice level.
 - i. In one example, one or more syntax elements may indicate whether to use the NN-based filter for the slice level.
 - ii. In one example, one or more syntax elements may indicate which NN-based filter is used.
 - iii. In one example, the syntax elements may indicate the usage of NN-based filter separately for Y, U, and V components for slice level.
 - 1) In one example, the syntax elements indicating the usage of NN-based filter separately for Y, U, and V components may be dependent on the syntax element which indicating whether to use the NN-based filter for slice level.
 - iv. In one example, one or more syntax elements may indicate whether NN-based filters can be adaptively selected in the CTU/CTB level.
 - b. In one example, one or more syntax elements may be signaled to control the usage of NN-based filter in the CTU/CTB level.
 - i. In one example, one or more syntax elements may indicate whether to use the NN-based filter for CTU/CTB level.
- 20
- 25
- 30

ii. In one example, one or more syntax elements may indicate which NN-based filter is used for CTU/CTB level.

1) In one example, the syntax elements indicating which NN-based filter is used may be dependent on the syntax element indicating whether to use the NN-based filter for CTU/CTB level.

iii. In one example, the syntax elements may indicate the usage of NN-based filter separately for Y, U, and V components for CTU/CTB level.

1) In one example, the syntax elements indicating the usage of NN-based filter separately for Y, U, and V components may be dependent on the syntax element which indicating whether to use the NN-based filter for CTU/CTB level.

a. In one example, the syntax elements which indicating the usage of NN-based filter separately for Y, U, and V components may be signaled when the syntax element which indicating whether to use the NN-based filter for CTU/CTB level has been parsed and NN-based filter is used for CTU/CTB level.

b. In one example, the syntax elements which indicating the usage of NN-based filter separately for luma and chroma components may be NOT signaled when the syntax element which indicating whether to use the NN-based filter for CTU/CTB level has been parsed and NN-based filter is NOT used for CTU/CTB level.

2) In one example, the syntax elements indicating the usage of NN-based filter for V component may be dependent on the syntax elements indicating the usage of NN-based filter for Y, and U components.

a. In one example, the syntax elements indicating the usage of NN-based filter for V component may be NOT signalled and inferred to use NN-based filter for V component when the syntax elements indicating the usage of NN-based filter for Y, and U components have been parsed and NN-based filter is NOT used for Y, and U component.

- b. In one example, the syntax elements indicating the usage of NN-based filter for V component may be signalled when the syntax elements indicating the usage of NN-based filter for Y, and U component has been parsed and NN-based filter is used for Y, and/or U component.

iv. In one example, the syntax elements may indicate the usage of NN-based filter separately for luma and chroma components for CTU/CTB level.

- 1) In one example, the syntax elements indicating the usage of NN-based filter separately for luma and chroma components may be dependent on the syntax element which indicating whether to use the NN-based filter for CTU/CTB level.

- a. In one example, the syntax elements which indicating the usage of NN-based filter separately for luma and chroma components may be signaled when the syntax element which indicating whether to use the NN-based filter for CTU/CTB level has been parsed and NN-based filter is used for CTU/CTB level.

- b. In one example, the syntax elements which indicating the usage of NN-based filter separately for luma and chroma components may be NOT signaled when the syntax element which indicating whether to use the NN-based filter for CTU/CTB level has been parsed and NN-based filter is NOT used for CTU/CTB level.

- 2) In one example, the syntax elements indicating the usage of NN-based filter for chroma component may be dependent on the syntax elements indicating the usage of NN-based filter for luma component.

- a. In one example, the syntax elements indicating the usage of NN-based filter for chroma component may be NOT signalled and inferred to use NN-based filter for chroma component when the syntax elements indicating the usage of NN-based filter for luma component has been parsed and NN-based filter is NOT used for luma component.

- b. In one example, the syntax elements indicating the usage of NN-based filter for chroma component may be signalled

when the syntax elements indicating the usage of NN-based filter for luma component has been parsed and NN-based filter is used for luma component.

3) In one example, the syntax elements may indicate the usage of NN-based filter separately for U, and V components in CTU/CTB level.

a. In one example, the syntax elements indicating the usage of NN-based filter separately for U, and V components may be dependent on the syntax element which indicating whether to use the NN-based filter for chroma component in CTU/CTB level.

i. In one example, the syntax elements which indicating the usage of NN-based filter separately for U, and V components may be signaled when the syntax element which indicating whether to use the NN-based filter for chroma component in CTU/CTB level has been parsed and NN-based filter is used for chroma component in CTU/CTB level.

ii. In one example, the syntax elements which indicating the usage of NN-based filter separately for U, and V components may be NOT signaled when the syntax element which indicating whether to use the NN-based filter for chroma component in CTU/CTB level has been parsed and NN-based filter is NOT used for chroma component in CTU/CTB level.

b. In one example, the syntax elements indicating the usage of NN-based filter for V component may be dependent on the syntax elements indicating the usage of NN-based filter for U components.

i. In one example, the syntax elements indicating the usage of NN-based filter for V component may be NOT signalled and inferred to use NN-based filter for V component when the syntax elements indicating the usage of NN-based filter for U components have been

parsed and NN-based filter is NOT used for U component.

- ii. In one example, the syntax elements indicating the usage of NN-based filter for V component may be signalled when the syntax elements indicating the usage of NN-based filter for U component has been parsed and NN-based filter is used for U component.

- v. In one example, whether to signal the syntax elements in CTU/CTB level may be dependent on the syntax elements signaled in the slice level.

15. To solve problem 9, a single NN-based filters may be used for generating filtered outputs for both intra and inter slices.

- a. In one example, intra slice may refer to I slice.

- i. In one example, inter slice may refer to B slice or P slice or B and P slices.

To design unified NN filter

16. It is proposed that a unified NN filter is designed by the all or part of mentioned items which should not be interpreted in a narrow way.

5. Embodiments

5.1 Embodiment 1

[0154] Fig. 17 illustrates an example network structure of NN filter. The Network structure of NN filter in Fig. 17 includes Item 3 side information and Item 4 multi-scale NN structure.

5.2 Embodiment 2

[0155] Fig. 18 illustrates an example network structure of NN filter. The Network structure of NN filter in Fig. 18 includes Item 1 adaptive channel number, Item 3 side information, and Item 4 multi-scale NN structure. The Conv A, Conv B, ..., Conv G are convolutions with different channel numbers. The larger text font means larger channel numbers, for example, the Conv A is convolution with the largest channel number.

5.3 Embodiment 3

[0156] Fig. 19 illustrates an example network structure of NN filter. The Network structure of NN filter in Fig. 19 includes Item 2 decomposition, Item 3 side information, and Item 4 multi-scale NN structure.

5.4 Embodiment 4

[0157] Fig. 20 illustrates an example network structure of NN filter. The Network structure of NN filter in Fig. 20 includes Item 1 adaptive channel number, Item 2 decomposition, Item 3

side information, and Item 4 multi-scale NN structure. The Conv A, Conv B, ..., Conv G are convolutions with different channel numbers. The larger text font means larger channel numbers, for example, the Conv A is convolution with the largest channel number.

5.5 Embodiment 5

5 [0158] Fig. 21 illustrates an example network structure of NN filter. The Network structure of NN filter of Fig. 21 includes Item 1 adaptive channel number, Item 2 decomposition, Item 3 side information, Item 4 multi-scale NN structure, and Item 5 Transformer.

5.6 Embodiment 6

10 [0159] Fig. 22 illustrates an example network structure of NN filter. The Network structure of NN filter in Fig. 22 includes Item 1 adaptive channel number, Item 2 decomposition, Item 3 side information, Item 4 multi-scale NN structure, and Item 5 Transformer. The Conv A, Conv B, ..., Conv G are convolutions with different channel numbers. The larger text font means larger channel numbers, for example, the Conv A is convolution with the largest channel number.

15 5.7 Embodiment 7

[0160] Separate models are used for generating luma and chroma components, respectively. Figs. 23A-23C illustrates an example structure and examples parameters of a luma model in accordance with some embodiments of the present disclosure. The Figs. 23A-23C show the structure and the number in the table shows the hyper-parameters of each convolution layer.

20 [0161] Figs. 24A-24C illustrates an example structure and examples parameters of a chroma model in accordance with some embodiments of the present disclosure. Figs. 24A-24C show the structure and the number in the table shows the hyper-parameters of each convolution layer.

5.8 Embodiment 8

25 [0162] One single model is used for generating the output of luma and chroma components together. Figs. 25A-25C illustrates an example structure and examples parameters of a single model for processing luma and chroma components in accordance with some embodiments of the present disclosure. Figs. 25A-25C show the structure and the number in the table shows the hyper-parameters of each convolution layer.

30 [0163] More details of the embodiments of the present disclosure will be described below which are related to neural network-based in-loop filtering for neural network-based video coding. As used herein, the term “video unit” may represent a block, a subblock, a coding tree block (CTB), a coding tree unit (CTU), a coding block (CB), a coding unit (CU), a prediction unit (PU), a transform unit (TU), a prediction block (PB),

a transform block (TB), a tile, a slice, a subpicture, a video processing unit comprising multiple samples/pixels, and/or the like. A video unit may be rectangular or non-rectangular. In addition, the term “neural network based (NN-based) filter” and “neural-network (NN) filter” may be used interchangeably. The embodiments of the present disclosure should be considered as examples to explain the general concepts and should not be interpreted in a narrow way. Furthermore, these embodiments can be applied individually or combined in any manner.

[0164] Fig. 26 illustrates a flowchart of a method 2600 for video processing in accordance with some embodiments of the present disclosure. As shown in Fig. 26, at 2602, a conversion between a current video unit of a video and a bitstream of the video is performed. In some embodiments, the conversion may include encoding the current video unit into the bitstream. Alternatively or additionally, the conversion may include decoding the current video unit from the bitstream.

[0165] The bitstream comprises at least one indication indicating first information regarding a usage of an NN-based filter for the current video unit. The NN-based filter is configured to process a luma component and a chroma component of the current video unit. In other words, the NN-based filter is capable of processing both luma and chroma components.

[0166] In some embodiments, the first information may comprise whether to use the NN-based filter for the current video unit. Additionally or alternatively, the first information may comprise at least one parameter for the NN-based filter that is used for the current video unit. For example, the at least one parameter may comprise a quantization parameter (QP) or the like. In some additional or alternative embodiments, the first information may comprise whether information regarding the usage of the NN-based filter is determined at a first level lower than a level of the current video unit. By way of example rather than limitation, the current video unit may be a slice, and thus the first level may be a level lower than the slice level, such as a coding tree unit (CTU) level, a coding tree block (CTB) level, or the like.

[0167] In some embodiments, the at least one indication may comprise a single indication. As used herein, an indication may be implemented as a syntax element, an index, a flag, or the like. By way of example, the value range of the single indication may be from 0 to 4. The value of the single indication being equal to 0 may indicate that the

NN-based filter is not used for the current video unit. The value of the single indication being equal to 1 may indicate that the NN-based filter is used for the current video unit and a base QP is used for the NN-based filter. The value of the single indication being equal to 2 may indicate that the NN-based filter is used for the current video unit and a base QP plus an offset is used for the NN-based filter. The value of the single indication being equal to 3 may indicate that the NN-based filter is used for the current video unit and a base QP minus an offset is used for the NN-based filter. The value of the single indication being equal to 4 may indicate that information regarding the usage of the NN-based filter is determined at the first level lower than the level of the current video unit.

10 **[0168]** It should be understood that the above examples are described merely for purpose of description. The scope of the present disclosure is not limited in this respect. For example, the at least one indication may also comprise more than one indication.

[0169] In some alternative or additional embodiments, the first information may comprise at least one of the following: which NN-based filter may be used for the current video unit, whether to use the NN-based filter for a Y component of the current video unit, whether to use the NN-based filter for a U component of the current video unit, or whether to use the NN-based filter for a V component of the current video unit. Furthermore, whether the bitstream comprises one or more indications indicating at least one of the following may be dependent on an indication indicating whether to use the NN-based filter for the current video unit: whether to use the NN-based filter for a Y component of the current video unit, whether to use the NN-based filter for a U component of the current video unit, or whether to use the NN-based filter for a V component of the current video unit.

25 **[0170]** In view of the above, the NN-based filter is capable of processing both luma and chroma components and information regarding the usage of this NN-based filter is signaled. Compared with the conventional solution, the proposed method can advantageously enable flexible usage of the NN-based filter. Thereby, the coding efficiency can be improved.

30 **[0171]** In some further embodiments, the current video unit may comprise a plurality of video blocks at the first level lower than the level of the current video unit. In addition, one or more indications indicate second information regarding a usage of the NN-based filter for a first video block of the plurality of video blocks. In one example embodiment,

the first level may be a CTU level, and the first video block may comprise a CTU. In another example embodiment, the first level may be a CTB level, and the video block may comprise a CTB.

[0172] In some embodiments, the second information may comprise whether to use the NN-based filter for the first video block. Additionally or alternatively, the second information may comprise one or more parameters for the NN-based filter that are used for the first video block. For example, the one or more parameters may comprise a quantization parameter (QP) or the like.

[0173] By way of example, the one or more indications may comprise a first indication indicating whether to use the NN-based filter for the first video block, and a second indication indicating the one or more parameters. By way of example, the value range of the first indication may be from 0 to 1, and the value range of the first indication may be from 0 to 2. The value of the first indication being equal to 0 may indicate that the NN-based filter is not used for the first video block. The value of the first indication being equal to 1 may indicate that the NN-based filter is used for the first video block. The value of the second indication being equal to 0 may indicate that a base QP is used for the NN-based filter. The value of the second indication being equal to 1 may indicate that a base QP plus an offset is used for the NN-based filter. The value of the second indication being equal to 2 may indicate that a base QP minus an offset is used for the NN-based filter.

[0174] It should be understood that the above examples are described merely for purpose of description. The scope of the present disclosure is not limited in this respect. For example, the one or more indications may also be implemented with a single indication.

[0175] Furthermore, whether the second indication is comprised in the bitstream may be dependent on the first indication. For example, if the first indication indicates that the NN-based filter is used for the first video block, the second indication may be comprised in the bitstream. Additionally, if the first indication indicates that the NN-based filter is not used for the first video block, the second indication may be absent from the bitstream.

[0176] In some additional embodiments, whether the one or more indications are comprised in the bitstream may be dependent on the at least one indication indicating the first information. For example, if the at least one indication indicates that information

regarding the usage of the NN-based filter is determined at the first level, the one or more indications may be comprised in the bitstream. Additionally or alternatively, if the at least one indication indicates that information regarding the usage of the NN-based filter is not determined at the first level, the one or more indications may be absent from the bitstream.

- 5 [0177] Thereby, the proposed method can advantageously support adaptive usage of the NN-based filter at a lower level, such a CTU level or a CTB level, and thus the coding efficiency can be improved.

[0178] In some additional or alternative embodiments, the second information may comprise at least one of the following: which NN-based filter is used for the first video
10 block, whether to use the NN-based filter for a Y component of the first video block, whether to use the NN-based filter for a U component of the first video block, or whether to use the NN-based filter for a V component of the first video block.

[0179] In some embodiments, whether the bitstream comprises a first set of indications indicating at least one of the following may be dependent on a first indication indicating
15 whether to use the NN-based filter for the first video block: whether to use the NN-based filter for the Y component of the first video block, whether to use the NN-based filter for the U component of the first video block, or whether to use the NN-based filter for the V component of the first video block. By way of example, if the first indication indicates that the NN-based filter is used for the first video block, the bitstream may comprise the
20 first set of indications. Additionally or alternatively, if the first indication indicates that the NN-based filter is not used for the first video block, the first set of indications may be absent from the bitstream.

[0180] In some embodiments, a second set of indications indicates whether to use the NN-based filter for the Y component of the first video block and whether to use the NN-
25 based filter for the U component of the first video block. A third indication indicates whether to use the NN-based filter for the V component of the first video block. Moreover, whether the third indication is comprised in the bitstream may be dependent on the second set of indications.

[0181] For example, if the second set of indications indicates that the NN-based filter
30 is not used for the Y component and the U component of the first video block, the third indication may be absent from the bitstream. In this case, the third indication may be inferred to indicate that the NN-based filter is used for the V component of the first video

block. Furthermore, if the second set of indications indicates that the NN-based filter is used for at least one the Y component or the U component of the first video block, the third indication may be comprised in the bitstream.

[0182] In some embodiments, the second information may comprise at least one of the following: whether to use the NN-based filter for a luma component of the first video block, or whether to use the NN-based filter for a chroma component of the first video block. In this case, whether the bitstream comprises a third set of indications indicating at least one of the following may be dependent on a first indication indicating whether to use the NN-based filter for the first video block: whether to use the NN-based filter for the luma component of the first video block, or whether to use the NN-based filter for the chroma component of the first video block.

[0183] For example, if the first indication indicates that the NN-based filter is used for the first video block, the bitstream may comprise the third set of indications. Additionally or alternatively, if the first indication indicates that the NN-based filter is not used for the first video block, the third set of indications may be absent from the bitstream.

[0184] In some embodiments, a fourth indication indicates whether to use the NN-based filter for the luma component of the first video block. A fifth indication indicates whether to use the NN-based filter for the chroma component of the first video block. Moreover, whether the fifth indication is comprised in the bitstream may be dependent on the fourth indication.

[0185] For example, if the fourth indication indicates that the NN-based filter is not used for the luma component of the first video block, the fifth indication may be absent from the bitstream. In this case, the fifth indication may be inferred to indicate that the NN-based filter is used for the chroma component of the first video block. Furthermore, if the fourth indication indicates that the NN-based filter is used for the luma component of the first video block, the fifth indication may be comprised the bitstream.

[0186] In some further embodiments, the second information may further comprise at least one of the following: whether to use the NN-based filter for a U component of the first video block, or whether to use the NN-based filter for a V component of the first video block. In this case, whether the bitstream comprises a fourth set of indications indicating at least one of the following may be dependent on a fifth indication indicates whether to use the NN-based filter for the chroma component of the first video block:

whether to use the NN-based filter for the U component of the first video block, or whether to use the NN-based filter for the V component of the first video block. For example, if the fifth indication indicates that the NN-based filter is used for the chroma component of the first video block, the fourth set of indications may be comprised in the bitstream.

5 Additionally, if the fifth indication indicates that the NN-based filter is not used for the chroma component of the first video block, the fourth set of indications may be absent from the bitstream.

[0187] In some embodiments, a third indication indicates whether to use the NN-based filter for the V component of the first video block. A sixth indication indicates whether
10 to use the NN-based filter for the U component of the first video block. Moreover, whether the third indication is comprised in the bitstream may be dependent on the sixth indication.

[0188] For example, if the sixth indication indicates that the NN-based filter is not used for the U component of the first video block, the third indication may be absent from the
15 bitstream. In this case, the third indication may be inferred to indicate that the NN-based filter is used for the V component of the first video block. Furthermore, if the sixth indication indicates that the NN-based filter is used for the U component of the first video block, the third indication may be comprised in the bitstream.

[0189] In view of the above, the proposed method can advantageously enable flexible
20 and efficient signaling of information on applying the NN-based filter on different color components, and thus the coding efficiency can be improved.

[0190] In some additional embodiments, at 2602, the NN-based filter is determined according to a rule. The rule indicates at least one of: different convolution types are assigned to different inputs of the NN-based filter, a convolution with kernel size is
25 decomposed into a combination of a plurality of convolutions with smaller kernel size, which side information to be used as an input of the NN-based filter, a multi-scale neural network structure is used in the NN-based filter, a transformer-based structure is used in the NN-based filter, a non-NN-based filter is combined with the NN-based filter, or a set of parameters of the NN-based filter is adaptive.

30 [0191] In addition, the NN-based filter is applied to a reconstruction of the current video unit, and the conversion is performed based on a result of applying the NN-based filter. Compared with the conventional solution where filters are selected directly, the filters can

be adaptively combined for the current video unit. In this way, the coding effectiveness and coding efficiency can be improved.

[0192] In some embodiments, a convolution sharing a same kernel size and different channel numbers is assigned for each input. In some embodiments, the channel numbers for each input are represented as $C_1, C_2, \dots, C_n$, where n is an integer which denotes the number of inputs.

[0193] In some embodiments, a constraint where the channel numbers for different inputs are different is added. For example, the constraint is that for indexes i and j , $C_i \neq C_j$, and where $1 \leq i, j \leq n$ that $C_i \neq C_j$.

[0194] In some embodiments, a constraint where at least two inputs have different channel numbers. For example, the constraint is that there is at least one pair of indexes i and j , $C_i \neq C_j$, and where $1 \leq i, j \leq n$.

[0195] In some embodiments, a convolution sharing a same channel numbers and different kernel sizes is assigned for each input. In some embodiments, a 1×1 kernel size is used for a convolution of a part of inputs and a $K \times K$ kernel size is used for a convolution of rest inputs, where K is an integer value greater than 1. In some other embodiments, different convolution channel numbers and different convolution kernel size are assigned for each input.

[0196] In some embodiments, a $C_1 \times C_2 \times K \times K$ convolution is decomposed into the combination of the plurality of convolutions with smaller kernel size. In this case, K represents an integer value greater than 1, C_1 and C_2 represent an input channel number and output channel number of the convolution, respectively.

[0197] In some embodiments, the input channel number and output channel number of the convolution are not changed, if the convolution is decomposed. In some embodiments, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_2 \times 1 \times K$ convolution followed by $C_1 \times C_2 \times K \times 1$ convolution. In some other embodiments, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_2 \times 1 \times K$ convolution followed by $C_1 \times C_2 \times K \times 1$ convolution, and an activation layer is placed after each convolution. In some further embodiments, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_2 \times K \times 1$ convolution followed by $C_1 \times C_2 \times 1 \times K$ convolution. In some other embodiments, the $C_1 \times C_2 \times K \times K$ convolution is

decomposed into a combination of $C_1 \times C_2 \times K \times 1$ convolution followed by $C_1 \times C_2 \times 1 \times K$ convolution, and an activation layer is placed after each convolution.

[0198] In some embodiments, the input channel number and output channel number of the convolution are changed, if the convolution is decomposed. In some embodiments, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_3 \times 1 \times K$ convolution followed by $C_3 \times C_2 \times K \times 1$ convolution, where C_3 is a positive integer different from C_2 . In some other embodiments, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_3 \times 1 \times K$ convolution followed by $C_3 \times C_2 \times K \times 1$ convolution, and an activation layer is placed after each convolution, where C_3 is a positive integer different with C_2 . In some further embodiments, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_3 \times K \times 1$ convolution followed by $C_3 \times C_2 \times 1 \times K$ convolution, where C_3 is a positive integer different from C_2 . In some other embodiments, the $C_1 \times C_2 \times K \times K$ convolution is decomposed into a combination of $C_1 \times C_3 \times K \times 1$ convolution followed by $C_3 \times C_2 \times 1 \times K$ convolution, and an activation layer is placed after each convolution, where C_3 is a positive integer different from C_2 .

[0199] In some embodiments, a portion of $K \times K$ convolutions are decomposed. In some other embodiments, all $K \times K$ convolutions are decomposed.

[0200] In some embodiments, the side information is used as an extra input of the NN-based filter. For example, a slice quantization parameters (QP) is used as the extra input of the NN-based filter. In some embodiments, the slice QP is normalized by a formula of $\text{sliceQP} \div \text{MAX_QP}$, wherein a value of MAX_QP is 63. In some other embodiments, the slice QP is first tiled or spanned into a 2-dimensional array with the same size as the current video unit to be filtered.

[0201] In some embodiments, a base QP is used as an extra input of the NN-based filter. In some embodiments, the base QP is normalized by a formula of $\text{baseQP} \div \text{MAX_QP}$, wherein a value of MAX_QP is 63. In some other embodiments, the base QP is first tiled or spanned into a 2-dimensional array with the same size as the current video unit to be filtered.

[0202] In some embodiments, a prediction picture is used as an extra input of the NN-based filter. In some other embodiments, a slice type is used as an extra input of the NN-

based filter. For example, the slice type is a binary value which indicates whether a picture to be filtered is intra slice. As another example, a slice type indicator is a first tiled or spanned into a 2-dimensional array with the same size as the current video unit to be filtered.

5 **[0203]** In some embodiments, IPB information of the current video unit to be filtered is used as an extra input of the NN-based filter. For example, the IPB information is derived from a slice type. In some embodiments, a value of IPB information is derived as follows: if the slice type is I slice, the value of IPB information is equal to a, if the slice type is B slice, the value of IPB information is equal to b, and if the slice type is P slice, the value
10 of IPB information is equal to c, wherein a, b and c are constants. In some embodiments, $b = c$. Alternatively, or in addition, $b = -a$, $c = -a$. Alternatively, or in addition, $a = 1$, $b = -1$, $c = -1$. Alternatively, or in addition, $a = 1$, $b = 0.5$, $c = 0.5$. In some embodiments, the IPB information is first tiled or spanned into a 2-dimensional array with the same size as the current video unit to be filtered.

15 **[0204]** In some embodiments, a boundary strength of the current video unit to be filtered is used as an extra input of the NN-based filter. In some embodiments, a combination of the side information is used as the extra input of the NN-based filter.

[0205] In some embodiments, convolutions for each input side information are performed separately and then all convolution results are concatenated with an output of
20 convolution of reconstruction picture. In some embodiments, a reconstruction picture and side information are concatenated and followed by a convolution.

[0206] In some embodiments, the multi-scale neural network structure comprises a neural network with two branches is used. In some embodiments, as shown in Fig. 16, one of the two branches comprises a $C_1 \times C_2 \times K_1 \times K_1$ convolution and an activation layer,
25 the other branch comprises $C_3 \times C_4 \times K_2 \times K_2$ convolution and an activation layer, and where K_i represents a kernel size where $1 \leq i \leq 4$, C_j is related to channel numbers, where $1 \leq j \leq 8$.

[0207] In some embodiments, the multi-scale neural network structure that comprises a plurality of branches, a first convolution is used in a first branch of the plurality of
30 branches and a second convolution is used in a second branch of the plurality of branches. In addition, at least one of the followings of the first convolution is different from that of the second convolution: a kernel size, a channel number, or a stride.

[0208] In some embodiments, at least one of: head, backbone or tail is determined by using a transformer network. In some other embodiments, the transformer-based structure is combined with convolutional neural network (CNN) in the NN-based filter. For example, a CNN module is followed by a transformer module in the NN-based filter. In
5 some further embodiments, the transformer-based structure and a CNN-based structure are alternative in the NN-based filter. For example, the NN-based filters are selected, if there are two neural network-based filters which comprise a CNN-based filter and a transformer-based filter.

[0209] In some embodiments, the non-NN-based filter is a deblocking filter (DBF). In
10 some other embodiments, the non-NN-based filter is a sample adaptive offset (SAO) filter. Alternatively, the non-NN-based filter is a combination of DBF and SAO filter.

[0210] In some embodiments, the NN-based filter and a DBF are combined. In some embodiments, an SAO filter is after the combination of the NN-based filter and the DBF. For example, the SAO filter is enabled or disabled by a syntax element.

15 [0211] In some embodiments, an adaptive loop filter (ALF) is used after an SAO filter. In some embodiments, the ALF is enabled or disabled by a syntax element.

[0212] In some embodiments, the NN-based filter and a combination of DBF and SAO filter are combined. For example, an ALF is used after the combination of the NN-based filter and the combination of DBF and SAO filter. In some embodiments, the ALF is
20 enabled or disabled by a syntax element.

[0213] In some embodiments, reconstruction samples of the non-NN-based filter and the NN-based filter are combined by a scaling factor. In some embodiments, the reconstruction samples are combined in slice level or block level. In some embodiments, the scaling factor is adaptive and is determined by video content. In some other
25 embodiments, the scaling factor is pre-defined.

[0214] In some embodiments, the scaling factor is separated for different components. For example, the scaling factor is separated for luma and chroma components. Alternatively, or in addition, the scaling factor is separated for U and V components.

[0215] In some embodiments, a candidate list comprising a plurality of input parameters
30 is used. In some embodiments, the number of candidates in the candidate list is configurable in sequence level or slice level or block level.

[0216] In some embodiments, the candidate list is constructed in sequence level or slice level or block level. In some other embodiments, the candidate list is adaptive for sequence level or slice level or block level.

[0217] In some embodiments, an input parameter is a variable dependent on base QP which is denoted as q . For example, q is QP in sequence level or slice level or block level.

[0218] In some embodiments, the candidate list comprises a base QP and adjusted QP. In some embodiments, the adjusted QP is equal to the base QP by adding an offset. In some embodiments, the offset is equal to one of -5, 10, 5. In some other embodiments, the offset is dependent on a thread identity (TID) level.

[0219] In some embodiments, an inference granularity or size of NN-based filter is adaptive for one of: sequence level, slice level, or block level. In some embodiments, the inference granularity or size is dependent on a slice type. In some other embodiments, the inference granularity or size is configurable dependent on a syntax element in one of: sequence level, slice level, or block level.

[0220] In some embodiments, at least one of: block extension or padding size is adaptive for one of: sequence level, slice level, or block level. In some embodiments, at least one of: block extension or padding size dependent on block modes and/or slice type. In some other embodiments, at least one of: block extension or padding size is configurable dependent on a syntax element in one of: sequence level, slice level, or block level.

[0221] In some embodiments, at 2602, the NN-based filter is determined. The NN-based filter comprises a multi-scale neural network structure that comprises a plurality of branches. Moreover, the NN-based filter is applied to the current video unit, and the conversion is performed based on the filtered video unit. Compared with the conventional solution where filters are selected directly, the filters can be adaptively combined for the current video unit. In this way, the coding effectiveness and coding efficiency can be improved.

[0222] In some embodiments, the NN-based filter comprises a combination of a plurality of convolutions with smaller kernel size. For example, the convolution with kernel size $K \times K$ is decomposed into the plurality of convolutions with smaller kernel size.

[0223] In some embodiments, a first convolution is used in a first branch of the plurality of branches and a second convolution is used in a second branch of the plurality of

branches. At least one of the followings of the first convolution may be different from that of the second convolution: a kernel size, a channel number, or a stride. In one example, neural network with multiple branches may be used. In each branch, different convolution with different kernel size, channel number, or stride might be used.

5 **[0224]** In some embodiments, an activation layer is used in the NN-based filter. In some embodiments, the activation layer is used after all convolution layers, or wherein the activation layer is used after a part of convolution layers. In some embodiments, the activation layer is a PRelu layer, or wherein the activation layer is a Relu layer. In some embodiments, a residual connection is used.

10 **[0225]** In some embodiments, the multi-scale neural network structure is used as a basic block in the NN-based filter. In some embodiments, the basic block is used for a predetermined number of times. The predetermined number may be an integer greater than 0.

15 **[0226]** In some embodiments, the basic block is used together with other blocks. In some embodiments, the basic block is used together with other convolution layer and activation layer, and/or wherein the basic block is used together with a residual block.

[0227] In some embodiments, the NN-based filter is used for at least one of: luma component or chroma component. Alternatively, or in addition, the NN-based filter is used for at least one of: intra slice or inter slice.

20 **[0228]** In some embodiments, an output of the NN-based filter comprises luma components, or chroma components. Alternatively, an output of the NN-based filter comprises both luma and chroma components. In some embodiments, a first NN-based filter is used for generating an output of luma component and a second NN-based filter is used for generating an output of chroma component, respectively.

25 **[0229]** In some embodiments, a first number of models are used for generating an output of luma component, and a second number of models are used for generating an output of chroma component. In this case, both the first number and the second number may be integers greater than 0. In some embodiments, Cb and Cr components share a same chroma model of the NN-based filter.

30 **[0230]** In some embodiments, a single NN-based filter is used for generating an output of luma component and an output of chroma component. In some embodiments, a third

number of models are used, and for each model, outputs of luma component and chroma component are generated, where M1 is an integer greater than 0.

[0231] In some embodiments, the outputs of luma and chroma components generated by the single NN-based filter are used together. In some other embodiments, the outputs of luma and chroma components generated by the single NN-based filter are used separately.

[0232] In some embodiments, the output of luma component generated by the single neural network is used and the output of chroma component generated by the NN-based filter is not used. In some embodiments, the output of luma component generated by the single neural network is not used and the output of chroma component generated by the NN-based filter is used.

[0233] In some embodiments, a single NN-based filter is used for generating filtered outputs for both intra and inter slices. In some embodiments, the intra slice is I slice. In some embodiments, the inter slice is one of: B slice, or P slice, or B and P slices.

[0234] In view of the above, the solutions in accordance with some embodiments of the present disclosure can advantageously improve coding efficiency and coding quality.

[0235] According to further embodiments of the present disclosure, a non-transitory computer-readable recording medium is provided. The non-transitory computer-readable recording medium stores a bitstream of a video which is generated by a method performed by an apparatus for video processing. In the method, a conversion between a current video unit of the video and the bitstream is performed. The bitstream comprises at least one indication indicating first information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit.

[0236] According to still further embodiments of the present disclosure, a method for storing bitstream of a video is provided. In the method, a conversion between a current video unit of the video and the bitstream is performed. The bitstream comprises at least one indication indicating first information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit. Furthermore, the bitstream is stored in a non-transitory computer-readable recording medium.

[0237] Implementations of the present disclosure can be described in view of the following clauses, the features of which can be combined in any reasonable manner.

[0238] Clause 1. A method for video processing, comprising: performing a conversion between a current video unit of a video and a bitstream of the video, wherein the bitstream
5 comprises at least one indication indicating first information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit.

[0239] Clause 2. The method of clause 1, wherein the current video unit comprises a slice.

10 [0240] Clause 3. The method of any of clauses 1-2, wherein the first information comprises at least one of the following: whether to use the NN-based filter for the current video unit, at least one parameter for the NN-based filter that is used for the current video unit, or whether information regarding the usage of the NN-based filter is determined at a first level lower than a level of the current video unit.

15 [0241] Clause 4. The method of clause 3, wherein the at least one parameter comprises a quantization parameter (QP).

[0242] Clause 5. The method of any of clauses 3-4, wherein the first level is a coding tree unit (CTU) level or a coding tree block (CTB) level.

20 [0243] Clause 6. The method of any of clauses 1-5, wherein the at least one indication comprises a single indication.

[0244] Clause 7. The method of any of clauses 1-6, wherein the current video unit comprises a plurality of video blocks at a first level lower than a level of the current video unit, and one or more indications indicate second information regarding a usage of the NN-based filter for a first video block of the plurality of video blocks.

25 [0245] Clause 8. The method of clause 7, wherein the second information comprises at least one of the following: whether to use the NN-based filter for the first video block, or one or more parameters for the NN-based filter that are used for the first video block.

[0246] Clause 9. The method of clause 8, wherein the one or more indications comprise: a first indication indicating whether to use the NN-based filter for the first video block,
30 and a second indication indicating the one or more parameters.

[0247] Clause 10. The method of clause 9, wherein whether the second indication is comprised in the bitstream is dependent on the first indication.

[0248] Clause 11. The method of clause 10, wherein if the first indication indicates that the NN-based filter is used for the first video block, the second indication is comprised in the bitstream.

[0249] Clause 12. The method of any of clauses 8-11, wherein the one or more parameters comprise a quantization parameter (QP).

[0250] Clause 13. The method of any of clauses 7-12, wherein the first level is a CTU level, and the first video block comprises a CTU, or wherein the first level is a CTB level, and the video block comprises a CTB.

[0251] Clause 14. The method of any of clauses 7-13, wherein whether the one or more indications are comprised in the bitstream is dependent on the at least one indication.

[0252] Clause 15. The method of clause 14, wherein if the at least one indication indicates that information regarding the usage of the NN-based filter is determined at the first level, the one or more indications are comprised in the bitstream.

[0253] Clause 16. The method of any of clauses 1-15, wherein the first information comprises at least one of the following: which NN-based filter is used for the current video unit, whether to use the NN-based filter for a Y component of the current video unit, whether to use the NN-based filter for a U component of the current video unit, or whether to use the NN-based filter for a V component of the current video unit.

[0254] Clause 17. The method of clause 16, wherein whether the bitstream comprises one or more indications indicating at least one of the following is dependent on an indication indicating whether to use the NN-based filter for the current video unit: whether to use the NN-based filter for a Y component of the current video unit, whether to use the NN-based filter for a U component of the current video unit, or whether to use the NN-based filter for a V component of the current video unit.

[0255] Clause 18. The method of any of clauses 7-17, wherein the second information comprises at least one of the following: which NN-based filter is used for the first video block, whether to use the NN-based filter for a Y component of the first video block, whether to use the NN-based filter for a U component of the first video block, or whether to use the NN-based filter for a V component of the first video block.

[0256] Clause 19. The method of clause 18, wherein whether the bitstream comprises a first set of indications indicating at least one of the following is dependent on a first indication indicating whether to use the NN-based filter for the first video block: whether to use the NN-based filter for the Y component of the first video block, whether to use the NN-based filter for the U component of the first video block, or whether to use the NN-based filter for the V component of the first video block.

[0257] Clause 20. The method of clause 19, wherein if the first indication indicates that the NN-based filter is used for the first video block, the bitstream comprises the first set of indications.

[0258] Clause 21. The method of any of clauses 19-20, wherein if the first indication indicates that the NN-based filter is not used for the first video block, the first set of indications is absent from the bitstream.

[0259] Clause 22. The method of any of clauses 18-21, wherein a second set of indications indicates whether to use the NN-based filter for the Y component of the first video block and whether to use the NN-based filter for the U component of the first video block, a third indication indicates whether to use the NN-based filter for the V component of the first video block, and whether the third indication is comprised in the bitstream is dependent on the second set of indications.

[0260] Clause 23. The method of clause 22, wherein if the second set of indications indicates that the NN-based filter is not used for the Y component and the U component of the first video block, the third indication is absent from the bitstream.

[0261] Clause 24. The method of any of clauses 22-23, wherein if the second set of indications indicates that the NN-based filter is used for at least one the Y component or the U component of the first video block, the third indication is comprised in the bitstream.

[0262] Clause 25. The method of any of clauses 7-24, wherein the second information comprises at least one of the following: whether to use the NN-based filter for a luma component of the first video block, or whether to use the NN-based filter for a chroma component of the first video block.

[0263] Clause 26. The method of clause 25, wherein whether the bitstream comprises a third set of indications indicating at least one of the following is dependent on a first indication indicating whether to use the NN-based filter for the first video block: whether

to use the NN-based filter for the luma component of the first video block, or whether to use the NN-based filter for the chroma component of the first video block.

[0264] Clause 27. The method of clause 26, wherein if the first indication indicates that the NN-based filter is used for the first video block, the bitstream comprises the third set of indications.

[0265] Clause 28. The method of any of clauses 26-27, wherein if the first indication indicates that the NN-based filter is not used for the first video block, the third set of indications is absent from the bitstream.

[0266] Clause 29. The method of any of clauses 25-28, wherein a fourth indication indicates whether to use the NN-based filter for the luma component of the first video block, a fifth indication indicates whether to use the NN-based filter for the chroma component of the first video block, and whether the fifth indication is comprised in the bitstream is dependent on the fourth indication.

[0267] Clause 30. The method of clause 29, wherein if the fourth indication indicates that the NN-based filter is not used for the luma component of the first video block, the fifth indication is absent from the bitstream.

[0268] Clause 31. The method of any of clauses 29-30, wherein if the fourth indication indicates that the NN-based filter is used for the luma component of the first video block, the fifth indication is comprised the bitstream.

[0269] Clause 32. The method of any of clauses 25-31, wherein the second information further comprises at least one of the following: whether to use the NN-based filter for a U component of the first video block, or whether to use the NN-based filter for a V component of the first video block.

[0270] Clause 33. The method of clause 32, wherein whether the bitstream comprises a fourth set of indications indicating at least one of the following is dependent on a fifth indication indicates whether to use the NN-based filter for the chroma component of the first video block: whether to use the NN-based filter for the U component of the first video block, or whether to use the NN-based filter for the V component of the first video block.

[0271] Clause 34. The method of clause 33, wherein if the fifth indication indicates that the NN-based filter is used for the chroma component of the first video block, the fourth set of indications is comprised in the bitstream.

[0272] Clause 35. The method of any of clauses 33-34, wherein if the fifth indication indicates that the NN-based filter is not used for the chroma component of the first video block, the fourth set of indications is absent from the bitstream.

[0273] Clause 36. The method of any of clauses 32-35, wherein a third indication indicates whether to use the NN-based filter for the V component of the first video block, a sixth indication indicates whether to use the NN-based filter for the U component of the first video block, and whether the third indication is comprised in the bitstream is dependent on the sixth indication.

[0274] Clause 37. The method of clause 36, wherein if the sixth indication indicates that the NN-based filter is not used for the U component of the first video block, the third indication is absent from the bitstream.

[0275] Clause 38. The method of any of clauses 36-37, wherein if the sixth indication indicates that the NN-based filter is used for the U component of the first video block, the third indication is comprised in the bitstream.

[0276] Clause 39. The method of any of clauses 1-38, wherein an indication comprises a syntax element.

[0277] Clause 40. The method of any of clauses 1-39, wherein performing the conversion comprises: determining the NN-based filter according to a rule, wherein the rule indicates at least one of: different convolution types are assigned to different inputs of the NN-based filter, a convolution with kernel size is decomposed into a combination of a plurality of convolutions with smaller kernel size, which side information to be used as an input of the NN-based filter, a multi-scale neural network structure is used in the NN-based filter, a transformer-based structure is used in the NN-based filter, a non-NN-based filter is combined with the NN-based filter, or a set of parameters of the NN-based filter is adaptive; applying the NN-based filter to a reconstruction of the current video unit; and performing the conversion based on the applying.

[0278] Clause 41. The method of clause 40, wherein a convolution sharing a same kernel size and different channel numbers is assigned for each input.

[0279] Clause 42. The method of clause 40, wherein a convolution sharing a same channel numbers and different kernel sizes is assigned for each input.

[0280] Clause 43. The method of clause 40, wherein different convolution channel

numbers and different convolution kernel size are assigned for each input.

[0281] Clause 44. The method of clause 40, wherein a $C_1 \times C_2 \times K \times K$ convolution is decomposed into the combination of the plurality of convolutions with smaller kernel size, wherein K represents an integer value greater than 1, C_1 and C_2 represent an input channel number and output channel number of the convolution, respectively.

[0282] Clause 45. The method of clause 44, wherein the input channel number and output channel number of the convolution are not changed, if the convolution is decomposed.

[0283] Clause 46. The method of clause 44, wherein the input channel number and output channel number of the convolution are changed, if the convolution is decomposed.

[0284] Clause 47. The method of clause 44, wherein a portion of $K \times K$ convolutions are decomposed.

[0285] Clause 48. The method of clause 44, wherein all $K \times K$ convolutions are decomposed.

[0286] Clause 49. The method of clause 40, wherein the side information is used as an extra input of the NN-based filter.

[0287] Clause 50. The method of clause 49, wherein a slice QP is used as the extra input of the NN-based filter.

[0288] Clause 51. The method of clause 49, wherein a base QP is used as an extra input of the NN-based filter.

[0289] Clause 52. The method of clause 49, wherein a prediction picture is used as an extra input of the NN-based filter.

[0290] Clause 53. The method of clause 49, wherein a slice type is used as an extra input of the NN-based filter.

[0291] Clause 54. The method of clause 49, wherein IPB information of the current video unit to be filtered is used as an extra input of the NN-based filter.

[0292] Clause 55. The method of clause 49, wherein a boundary strength of the current video unit to be filtered is used as an extra input of the NN-based filter.

[0293] Clause 56. The method of any of clauses 49-55, wherein a combination of the

side information is used as the extra input of the NN-based filter.

[0294] Clause 57. The method of clause 40, wherein convolutions for each input side information are performed separately and then all convolution results are concatenated with an output of convolution of reconstruction picture.

5 **[0295]** Clause 58. The method of clause 40, wherein a reconstruction picture and side information are concatenated and followed by a convolution.

[0296] Clause 59. The method of clause 40, wherein the multi-scale neural network structure comprises a neural network with two branches is used, one of the two branches comprises a $C_1 \times C_2 \times K_1 \times K_1$ convolution and an activation layer, the other branch
10 comprises $C_3 \times C_4 \times K_2 \times K_2$ convolution and an activation layer, and wherein K_i represents a kernel size where $1 \leq i \leq 4$, C_j is related to channel numbers, where $1 \leq j \leq 8$.

[0297] Clause 60. The method of clause 40, wherein the multi-scale neural network structure that comprises a plurality of branches, a first convolution is used in a first branch
15 of the plurality of branches and a second convolution is used in a second branch of the plurality of branches, and at least one of the followings of the first convolution is different from that of the second convolution: a kernel size, a channel number, or a stride.

[0298] Clause 61. The method of clause 40, wherein at least one of: head, backbone or tail is determined by using a transformer network.

20 **[0299]** Clause 62. The method of clause 40, wherein the transformer-based structure is combined with convolutional neural network (CNN) in the NN-based filter.

[0300] Clause 63. The method of clause 62, wherein a CNN module is followed by a transformer module in the NN-based filter.

[0301] Clause 64. The method of clause 40, wherein the transformer-based structure
25 and a CNN-based structure are alternative in the NN-based filter.

[0302] Clause 65. The method of clause 64, wherein the NN-based filter is selected, if there are two neural network-based filters which comprise a CNN-based filter and a transformer-based filter.

[0303] Clause 66. The method of clause 40, wherein the non-NN-based filter is a
30 deblocking filter (DBF), or wherein the non-NN-based filter is a sample adaptive offset

(SAO) filter, or wherein the non-NN-based filter is a combination of DBF and SAO filter.

[0304] Clause 67. The method of clause 40, wherein the NN-based filter and a DBF are combined.

[0305] Clause 68. The method of clause 40, wherein the NN-based filter and a
5 combination of DBF and SAO filter are combined.

[0306] Clause 69. The method of clause 40, wherein reconstruction samples of the non-NN-based filter and the NN-based filter are combined by a scaling factor.

[0307] Clause 70. The method of clause 69, wherein the reconstruction samples are combined in slice level or block level.

10 **[0308]** Clause 71. The method of clause 69, wherein the scaling factor is adaptive and is determined by video content.

[0309] Clause 72. The method of clause 69, wherein the scaling factor is pre-defined.

[0310] Clause 73. The method of clause 69, wherein the scaling factor is separated for different components.

15 **[0311]** Clause 74. The method of clause 40, wherein a candidate list comprising a plurality of input parameters is used.

[0312] Clause 75. The method of clause 74, wherein the number of candidates in the candidate list is configurable in sequence level or slice level or block level.

[0313] Clause 76. The method of clause 74, wherein the candidate list is constructed in
20 sequence level or slice level or block level.

[0314] Clause 77. The method of clause 74, wherein the candidate list is adaptive for sequence level or slice level or block level.

[0315] Clause 78. The method of clause 74, wherein an input parameter is a variable dependent on base QP which is denoted as q .

25 **[0316]** Clause 79. The method of clause 74, wherein the candidate list comprises a base QP and adjusted QP.

[0317] Clause 80. The method of clause 40, wherein an inference granularity or size of NN-based filter is adaptive for one of: sequence level, slice level, or block level.

[0318] Clause 81. The method of clause 80, wherein the inference granularity or size is dependent on a slice type.

[0319] Clause 82. The method of clause 80, wherein the inference granularity or size is configurable dependent on a syntax element in one of: sequence level, slice level, or block level.

[0320] Clause 83. The method of clause 40, wherein at least one of: block extension or padding size is adaptive for one of: sequence level, slice level, or block level.

[0321] Clause 84. The method of clause 80, wherein at least one of: block extension or padding size dependent on block modes and/or slice type.

10 **[0322]** Clause 85. The method of clause 80, wherein at least one of: block extension or padding size is configurable dependent on a syntax element in one of: sequence level, slice level, or block level.

[0323] Clause 86. The method of any of clauses 1-85, wherein the NN-based filter is used for generating filtered outputs for both intra and inter slices.

15 **[0324]** Clause 87. The method of clause 86, wherein the intra slice is I slice.

[0325] Clause 88. The method of clause 86, wherein the inter slice is one of: B slice, or P slice, or B and P slices.

[0326] Clause 89. The method of any of clauses 1-88, wherein the conversion includes encoding the current video unit into the bitstream.

20 **[0327]** Clause 90. The method of any of clauses 1-88, wherein the conversion includes decoding the current video unit from the bitstream.

[0328] Clause 91. An apparatus for video processing comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform a method in accordance with any of clauses
25 1-90.

[0329] Clause 92. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of clauses 1-90.

[0330] Clause 93. A non-transitory computer-readable recording medium storing a

bitstream of a video which is generated by a method performed by an apparatus for video processing, wherein the method comprises: performing a conversion between a current video unit of the video and the bitstream, wherein the bitstream comprises at least one indication indicating first information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit.

[0331] Clause 94. A method for storing a bitstream of a video, comprising: performing a conversion between a current video unit of the video and the bitstream, wherein the bitstream comprises at least one indication indicating first information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit; and storing the bitstream in a non-transitory computer-readable recording medium.

Example Device

[0332] Fig. 27 illustrates a block diagram of a computing device 2700 in which various embodiments of the present disclosure can be implemented. The computing device 2700 may be implemented as or included in the source device 110 (or the video encoder 114 or 200) or the destination device 120 (or the video decoder 124 or 300).

[0333] It would be appreciated that the computing device 2700 shown in Fig. 27 is merely for purpose of illustration, without suggesting any limitation to the functions and scopes of the embodiments of the present disclosure in any manner.

[0334] As shown in Fig. 27, the computing device 2700 includes a general-purpose computing device 2700. The computing device 2700 may at least comprise one or more processors or processing units 2710, a memory 2720, a storage unit 2730, one or more communication units 2740, one or more input devices 2750, and one or more output devices 2760.

[0335] In some embodiments, the computing device 2700 may be implemented as any user terminal or server terminal having the computing capability. The server terminal may be a server, a large-scale computing device or the like that is provided by a service provider. The user terminal may for example be any type of mobile terminal, fixed terminal, or portable terminal, including a mobile phone, station, unit, device, multimedia computer, multimedia tablet, Internet node, communicator, desktop computer, laptop

computer, notebook computer, netbook computer, tablet computer, personal communication system (PCS) device, personal navigation device, personal digital assistant (PDA), audio/video player, digital camera/video camera, positioning device, television receiver, radio broadcast receiver, E-book device, gaming device, or any combination thereof, including the accessories and peripherals of these devices, or any combination thereof. It would be contemplated that the computing device 2700 can support any type of interface to a user (such as “wearable” circuitry and the like).

[0336] The processing unit 2710 may be a physical or virtual processor and can implement various processes based on programs stored in the memory 2720. In a multi-processor system, multiple processing units execute computer executable instructions in parallel so as to improve the parallel processing capability of the computing device 2700. The processing unit 2710 may also be referred to as a central processing unit (CPU), a microprocessor, a controller or a microcontroller.

[0337] The computing device 2700 typically includes various computer storage medium. Such medium can be any medium accessible by the computing device 2700, including, but not limited to, volatile and non-volatile medium, or detachable and non-detachable medium. The memory 2720 can be a volatile memory (for example, a register, cache, Random Access Memory (RAM)), a non-volatile memory (such as a Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), or a flash memory), or any combination thereof. The storage unit 2730 may be any detachable or non-detachable medium and may include a machine-readable medium such as a memory, flash memory drive, magnetic disk or another other media, which can be used for storing information and/or data and can be accessed in the computing device 2700.

[0338] The computing device 2700 may further include additional detachable/non-detachable, volatile/non-volatile memory medium. Although not shown in Fig. 27, it is possible to provide a magnetic disk drive for reading from and/or writing into a detachable and non-volatile magnetic disk and an optical disk drive for reading from and/or writing into a detachable non-volatile optical disk. In such cases, each drive may be connected to a bus (not shown) via one or more data medium interfaces.

[0339] The communication unit 2740 communicates with a further computing device via the communication medium. In addition, the functions of the components in the computing device 2700 can be implemented by a single computing cluster or multiple

computing machines that can communicate via communication connections. Therefore, the computing device 2700 can operate in a networked environment using a logical connection with one or more other servers, networked personal computers (PCs) or further general network nodes.

- 5 **[0340]** The input device 2750 may be one or more of a variety of input devices, such as a mouse, keyboard, tracking ball, voice-input device, and the like. The output device 2760 may be one or more of a variety of output devices, such as a display, loudspeaker, printer, and the like. By means of the communication unit 2740, the computing device 2700 can further communicate with one or more external devices (not shown) such as the storage
- 10 devices and display device, with one or more devices enabling the user to interact with the computing device 2700, or any devices (such as a network card, a modem and the like) enabling the computing device 2700 to communicate with one or more other computing devices, if required. Such communication can be performed via input/output (I/O) interfaces (not shown).
- 15 **[0341]** In some embodiments, instead of being integrated in a single device, some or all components of the computing device 2700 may also be arranged in cloud computing architecture. In the cloud computing architecture, the components may be provided remotely and work together to implement the functionalities described in the present disclosure. In some embodiments, cloud computing provides computing, software, data
- 20 access and storage service, which will not require end users to be aware of the physical locations or configurations of the systems or hardware providing these services. In various embodiments, the cloud computing provides the services via a wide area network (such as Internet) using suitable protocols. For example, a cloud computing provider provides applications over the wide area network, which can be accessed through a web browser or
- 25 any other computing components. The software or components of the cloud computing architecture and corresponding data may be stored on a server at a remote position. The computing resources in the cloud computing environment may be merged or distributed at locations in a remote data center. Cloud computing infrastructures may provide the services through a shared data center, though they behave as a single access point for the
- 30 users. Therefore, the cloud computing architectures may be used to provide the components and functionalities described herein from a service provider at a remote location. Alternatively, they may be provided from a conventional server or installed directly or otherwise on a client device.

[0342] The computing device 2700 may be used to implement video encoding/decoding in embodiments of the present disclosure. The memory 2720 may include one or more video coding modules 2725 having one or more program instructions. These modules are accessible and executable by the processing unit 2710 to perform the functionalities of the various embodiments described herein.

[0343] In the example embodiments of performing video encoding, the input device 2750 may receive video data as an input 2770 to be encoded. The video data may be processed, for example, by the video coding module 2725, to generate an encoded bitstream. The encoded bitstream may be provided via the output device 2760 as an output 2780.

[0344] In the example embodiments of performing video decoding, the input device 2750 may receive an encoded bitstream as the input 2770. The encoded bitstream may be processed, for example, by the video coding module 2725, to generate decoded video data. The decoded video data may be provided via the output device 2760 as the output 2780.

[0345] While this disclosure has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present application as defined by the appended claims. Such variations are intended to be covered by the scope of this present application. As such, the foregoing description of embodiments of the present application is not intended to be limiting.

I/We Claim:

1. A method for video processing, comprising:

performing a conversion between a current video unit of a video and a bitstream of the
5 video, wherein the bitstream comprises at least one indication indicating first information
regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-
based filter is configured to process a luma component and a chroma component of the current
video unit.

10 2. The method of claim 1, wherein the current video unit comprises a slice.

3. The method of any of claims 1-2, wherein the first information comprises at least one
of the following:

whether to use the NN-based filter for the current video unit,

15 at least one parameter for the NN-based filter that is used for the current video unit, or

whether information regarding the usage of the NN-based filter is determined at a first
level lower than a level of the current video unit.

20 4. The method of claim 3, wherein the at least one parameter comprises a quantization
parameter (QP).

5. The method of any of claims 3-4, wherein the first level is a coding tree unit (CTU)
level or a coding tree block (CTB) level.

25 6. The method of any of claims 1-5, wherein the at least one indication comprises a
single indication.

7. The method of any of claims 1-6, wherein the current video unit comprises a plurality
of video blocks at a first level lower than a level of the current video unit, and one or more
30 indications indicate second information regarding a usage of the NN-based filter for a first video
block of the plurality of video blocks.

8. The method of claim 7, wherein the second information comprises at least one of the following:

whether to use the NN-based filter for the first video block, or
one or more parameters for the NN-based filter that are used for the first video block.

9. The method of claim 8, wherein the one or more indications comprise:
a first indication indicating whether to use the NN-based filter for the first video block,
and
a second indication indicating the one or more parameters.

10. The method of claim 9, wherein whether the second indication is comprised in the bitstream is dependent on the first indication.

11. The method of claim 10, wherein if the first indication indicates that the NN-based filter is used for the first video block, the second indication is comprised in the bitstream.

12. The method of any of claims 8-11, wherein the one or more parameters comprise a quantization parameter (QP).

13. The method of any of claims 7-12, wherein the first level is a CTU level, and the first video block comprises a CTU, or
wherein the first level is a CTB level, and the video block comprises a CTB.

14. The method of any of claims 7-13, wherein whether the one or more indications are comprised in the bitstream is dependent on the at least one indication.

15. The method of claim 14, wherein if the at least one indication indicates that information regarding the usage of the NN-based filter is determined at the first level, the one or more indications are comprised in the bitstream.

16. The method of any of claims 1-15, wherein the first information comprises at least one of the following:

which NN-based filter is used for the current video unit,
whether to use the NN-based filter for a Y component of the current video unit,

whether to use the NN-based filter for a U component of the current video unit, or
whether to use the NN-based filter for a V component of the current video unit.

17. The method of claim 16, wherein whether the bitstream comprises one or more
indications indicating at least one of the following is dependent on an indication indicating
whether to use the NN-based filter for the current video unit:

whether to use the NN-based filter for a Y component of the current video unit,
whether to use the NN-based filter for a U component of the current video unit, or
whether to use the NN-based filter for a V component of the current video unit.

18. The method of any of claims 7-17, wherein the second information comprises at
least one of the following:

which NN-based filter is used for the first video block,
whether to use the NN-based filter for a Y component of the first video block,
whether to use the NN-based filter for a U component of the first video block, or
whether to use the NN-based filter for a V component of the first video block.

19. The method of claim 18, wherein whether the bitstream comprises a first set of
indications indicating at least one of the following is dependent on a first indication indicating
whether to use the NN-based filter for the first video block:

whether to use the NN-based filter for the Y component of the first video block,
whether to use the NN-based filter for the U component of the first video block, or
whether to use the NN-based filter for the V component of the first video block.

20. The method of claim 19, wherein if the first indication indicates that the NN-based
filter is used for the first video block, the bitstream comprises the first set of indications.

21. The method of any of claims 19-20, wherein if the first indication indicates that the
NN-based filter is not used for the first video block, the first set of indications is absent from
the bitstream.

22. The method of any of claims 18-21, wherein a second set of indications indicates
whether to use the NN-based filter for the Y component of the first video block and whether to
use the NN-based filter for the U component of the first video block, a third indication indicates

whether to use the NN-based filter for the V component of the first video block, and whether the third indication is comprised in the bitstream is dependent on the second set of indications.

23. The method of claim 22, wherein if the second set of indications indicates that the NN-based filter is not used for the Y component and the U component of the first video block, the third indication is absent from the bitstream.

24. The method of any of claims 22-23, wherein if the second set of indications indicates that the NN-based filter is used for at least one the Y component or the U component of the first video block, the third indication is comprised in the bitstream.

25. The method of any of claims 7-24, wherein the second information comprises at least one of the following:

whether to use the NN-based filter for a luma component of the first video block, or whether to use the NN-based filter for a chroma component of the first video block.

26. The method of claim 25, wherein whether the bitstream comprises a third set of indications indicating at least one of the following is dependent on a first indication indicating whether to use the NN-based filter for the first video block:

whether to use the NN-based filter for the luma component of the first video block, or whether to use the NN-based filter for the chroma component of the first video block.

27. The method of claim 26, wherein if the first indication indicates that the NN-based filter is used for the first video block, the bitstream comprises the third set of indications.

28. The method of any of claims 26-27, wherein if the first indication indicates that the NN-based filter is not used for the first video block, the third set of indications is absent from the bitstream.

29. The method of any of claims 25-28, wherein a fourth indication indicates whether to use the NN-based filter for the luma component of the first video block, a fifth indication indicates whether to use the NN-based filter for the chroma component of the first video block, and whether the fifth indication is comprised in the bitstream is dependent on the fourth indication.

30. The method of claim 29, wherein if the fourth indication indicates that the NN-based filter is not used for the luma component of the first video block, the fifth indication is absent from the bitstream.

5

31. The method of any of claims 29-30, wherein if the fourth indication indicates that the NN-based filter is used for the luma component of the first video block, the fifth indication is comprised in the bitstream.

10

32. The method of any of claims 25-31, wherein the second information further comprises at least one of the following:

whether to use the NN-based filter for a U component of the first video block, or
whether to use the NN-based filter for a V component of the first video block.

15

33. The method of claim 32, wherein whether the bitstream comprises a fourth set of indications indicating at least one of the following is dependent on a fifth indication indicating whether to use the NN-based filter for the chroma component of the first video block:

whether to use the NN-based filter for the U component of the first video block, or
whether to use the NN-based filter for the V component of the first video block.

20

34. The method of claim 33, wherein if the fifth indication indicates that the NN-based filter is used for the chroma component of the first video block, the fourth set of indications is comprised in the bitstream.

25

35. The method of any of claims 33-34, wherein if the fifth indication indicates that the NN-based filter is not used for the chroma component of the first video block, the fourth set of indications is absent from the bitstream.

30

36. The method of any of claims 32-35, wherein a third indication indicates whether to use the NN-based filter for the V component of the first video block, a sixth indication indicates whether to use the NN-based filter for the U component of the first video block, and whether the third indication is comprised in the bitstream is dependent on the sixth indication.

37. The method of claim 36, wherein if the sixth indication indicates that the NN-based filter is not used for the U component of the first video block, the third indication is absent from the bitstream.

5 38. The method of any of claims 36-37, wherein if the sixth indication indicates that the NN-based filter is used for the U component of the first video block, the third indication is comprised in the bitstream.

10 39. The method of any of claims 1-38, wherein an indication comprises a syntax element.

40. The method of any of claims 1-39, wherein performing the conversion comprises: determining the NN-based filter according to a rule, wherein the rule indicates at least one of:

15 different convolution types are assigned to different inputs of the NN-based filter,
a convolution with kernel size is decomposed into a combination of a plurality of convolutions with smaller kernel size,
which side information to be used as an input of the NN-based filter,
a multi-scale neural network structure is used in the NN-based filter,
a transformer-based structure is used in the NN-based filter,
20 a non-NN-based filter is combined with the NN-based filter, or
a set of parameters of the NN-based filter is adaptive;
applying the NN-based filter to a reconstruction of the current video unit; and
performing the conversion based on the applying.

25 41. The method of claim 40, wherein a convolution sharing a same kernel size and different channel numbers is assigned for each input.

42. The method of claim 40, wherein a convolution sharing a same channel numbers and different kernel sizes is assigned for each input.

30 43. The method of claim 40, wherein different convolution channel numbers and different convolution kernel size are assigned for each input.

44. The method of claim 40, wherein a $C_1 \times C_2 \times K \times K$ convolution is decomposed

into the combination of the plurality of convolutions with smaller kernel size, wherein K represents an integer value greater than 1, C_1 and C_2 represent an input channel number and output channel number of the convolution, respectively.

5 45. The method of claim 44, wherein the input channel number and output channel number of the convolution are not changed, if the convolution is decomposed.

 46. The method of claim 44, wherein the input channel number and output channel number of the convolution are changed, if the convolution is decomposed.
10

 47. The method of claim 44, wherein a portion of $K \times K$ convolutions are decomposed.

 48. The method of claim 44, wherein all $K \times K$ convolutions are decomposed.

15 49. The method of claim 40, wherein the side information is used as an extra input of the NN-based filter.

 50. The method of claim 49, wherein a slice QP is used as the extra input of the NN-based filter.
20

 51. The method of claim 49, wherein a base QP is used as an extra input of the NN-based filter.

 52. The method of claim 49, wherein a prediction picture is used as an extra input of the
25 NN-based filter.

 53. The method of claim 49, wherein a slice type is used as an extra input of the NN-based filter.

30 54. The method of claim 49, wherein IPB information of the current video unit to be filtered is used as an extra input of the NN-based filter.

 55. The method of claim 49, wherein a boundary strength of the current video unit to be filtered is used as an extra input of the NN-based filter.

56. The method of any of claims 49-55, wherein a combination of the side information is used as the extra input of the NN-based filter.

5 57. The method of claim 40, wherein convolutions for each input side information are performed separately and then all convolution results are concatenated with an output of convolution of reconstruction picture.

10 58. The method of claim 40, wherein a reconstruction picture and side information are concatenated and followed by a convolution.

59. The method of claim 40, wherein the multi-scale neural network structure comprises a neural network with two branches is used, one of the two branches comprises a $C_1 \times C_2 \times K_1 \times K_1$ convolution and an activation layer, the other branch comprises
15 $C_3 \times C_4 \times K_2 \times K_2$ convolution and an activation layer, and wherein K_i represents a kernel size where $1 \leq i \leq 4$, C_j is related to channel numbers, where $1 \leq j \leq 8$.

20 60. The method of claim 40, wherein the multi-scale neural network structure that comprises a plurality of branches, a first convolution is used in a first branch of the plurality of branches and a second convolution is used in a second branch of the plurality of branches, and at least one of the followings of the first convolution is different from that of the second convolution: a kernel size, a channel number, or a stride.

25 61. The method of claim 40, wherein at least one of: head, backbone or tail is determined by using a transformer network.

62. The method of claim 40, wherein the transformer-based structure is combined with convolutional neural network (CNN) in the NN-based filter.

30 63. The method of claim 62, wherein a CNN module is followed by a transformer module in the NN-based filter.

64. The method of claim 40, wherein the transformer-based structure and a CNN-based structure are alternative in the NN-based filter.

65. The method of claim 64, wherein the NN-based filter is selected, if there are two neural network-based filters which comprise a CNN-based filter and a transformer-based filter.

5 66. The method of claim 40, wherein the non-NN-based filter is a deblocking filter (DBF), or

wherein the non-NN-based filter is a sample adaptive offset (SAO) filter, or
wherein the non-NN-based filter is a combination of DBF and SAO filter.

10 67. The method of claim 40, wherein the NN-based filter and a DBF are combined.

68. The method of claim 40, wherein the NN-based filter and a combination of DBF and SAO filter are combined.

15 69. The method of claim 40, wherein reconstruction samples of the non-NN-based filter and the NN-based filter are combined by a scaling factor.

70. The method of claim 69, wherein the reconstruction samples are combined in slice level or block level.

20 71. The method of claim 69, wherein the scaling factor is adaptive and is determined by video content.

72. The method of claim 69, wherein the scaling factor is pre-defined.

25 73. The method of claim 69, wherein the scaling factor is separated for different components.

30 74. The method of claim 40, wherein a candidate list comprising a plurality of input parameters is used.

75. The method of claim 74, wherein the number of candidates in the candidate list is configurable in sequence level or slice level or block level.

76. The method of claim 74, wherein the candidate list is constructed in sequence level or slice level or block level.

77. The method of claim 74, wherein the candidate list is adaptive for sequence level or slice level or block level.

78. The method of claim 74, wherein an input parameter is a variable dependent on base QP which is denoted as q .

79. The method of claim 74, wherein the candidate list comprises a base QP and adjusted QP.

80. The method of claim 40, wherein an inference granularity or size of NN-based filter is adaptive for one of: sequence level, slice level, or block level.

81. The method of claim 80, wherein the inference granularity or size is dependent on a slice type.

82. The method of claim 80, wherein the inference granularity or size is configurable dependent on a syntax element in one of: sequence level, slice level, or block level.

83. The method of claim 40, wherein at least one of: block extension or padding size is adaptive for one of: sequence level, slice level, or block level.

84. The method of claim 80, wherein at least one of: block extension or padding size dependent on block modes and/or slice type.

85. The method of claim 80, wherein at least one of: block extension or padding size is configurable dependent on a syntax element in one of: sequence level, slice level, or block level.

86. The method of any of claims 1-85, wherein the NN-based filter is used for generating filtered outputs for both intra and inter slices.

87. The method of claim 86, wherein the intra slice is I slice.

88. The method of claim 86, wherein the inter slice is one of: B slice, or P slice, or B and P slices.

5 89. The method of any of claims 1-88, wherein the conversion includes encoding the current video unit into the bitstream.

90. The method of any of claims 1-88, wherein the conversion includes decoding the current video unit from the bitstream.
10

91. An apparatus for video processing comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform a method in accordance with any of claims 1-90.

15 92. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of claims 1-90.

93. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by an apparatus for video processing, wherein the method comprises:
20

performing a conversion between a current video unit of the video and the bitstream, wherein the bitstream comprises at least one indication indicating first information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit.
25

94. A method for storing a bitstream of a video, comprising:
performing a conversion between a current video unit of the video and the bitstream, wherein the bitstream comprises at least one indication indicating first information regarding a usage of a neural network (NN) based filter for the current video unit, and the NN-based filter is configured to process a luma component and a chroma component of the current video unit;
30 and
storing the bitstream in a non-transitory computer-readable recording medium.

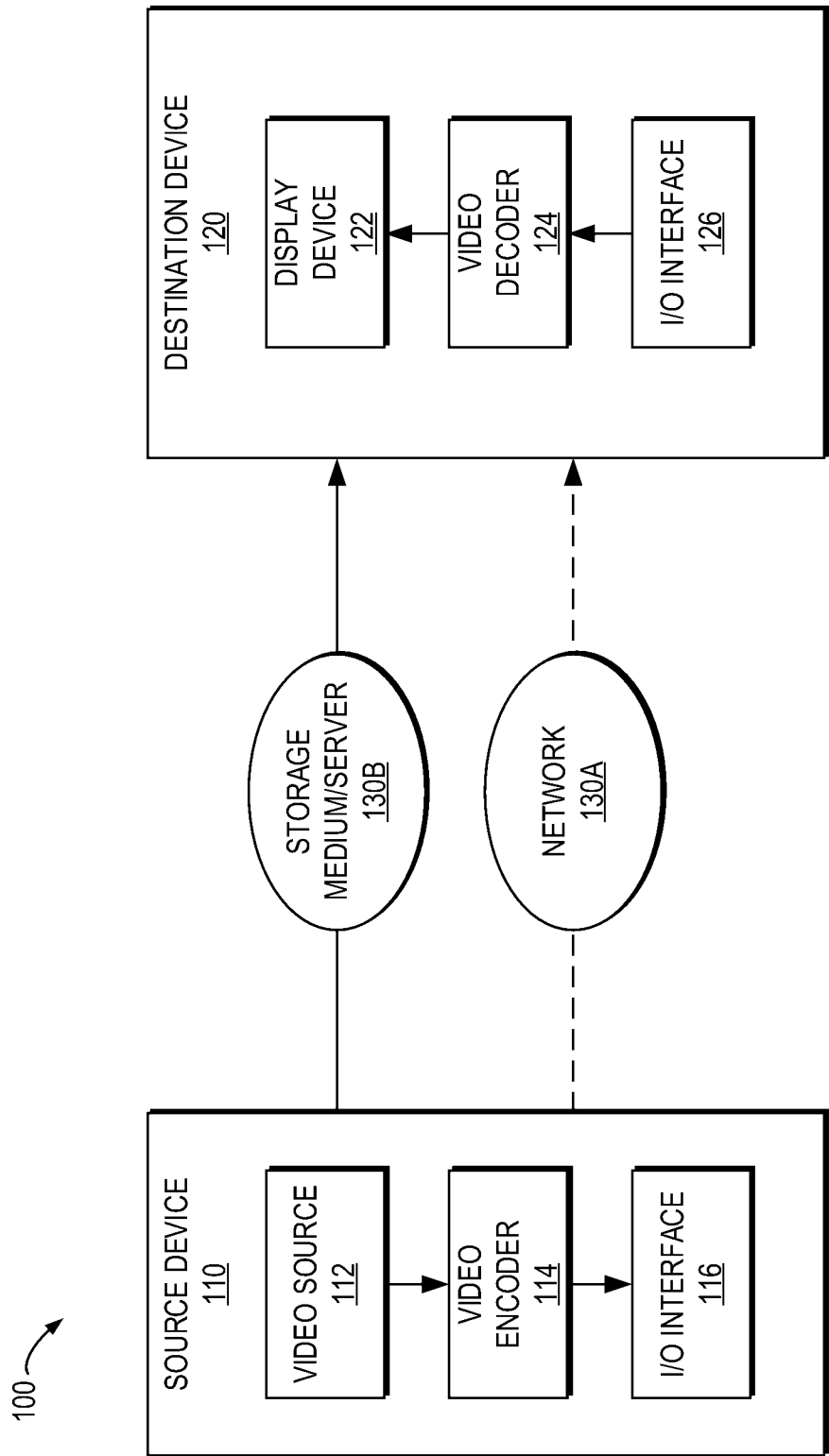


Fig. 1

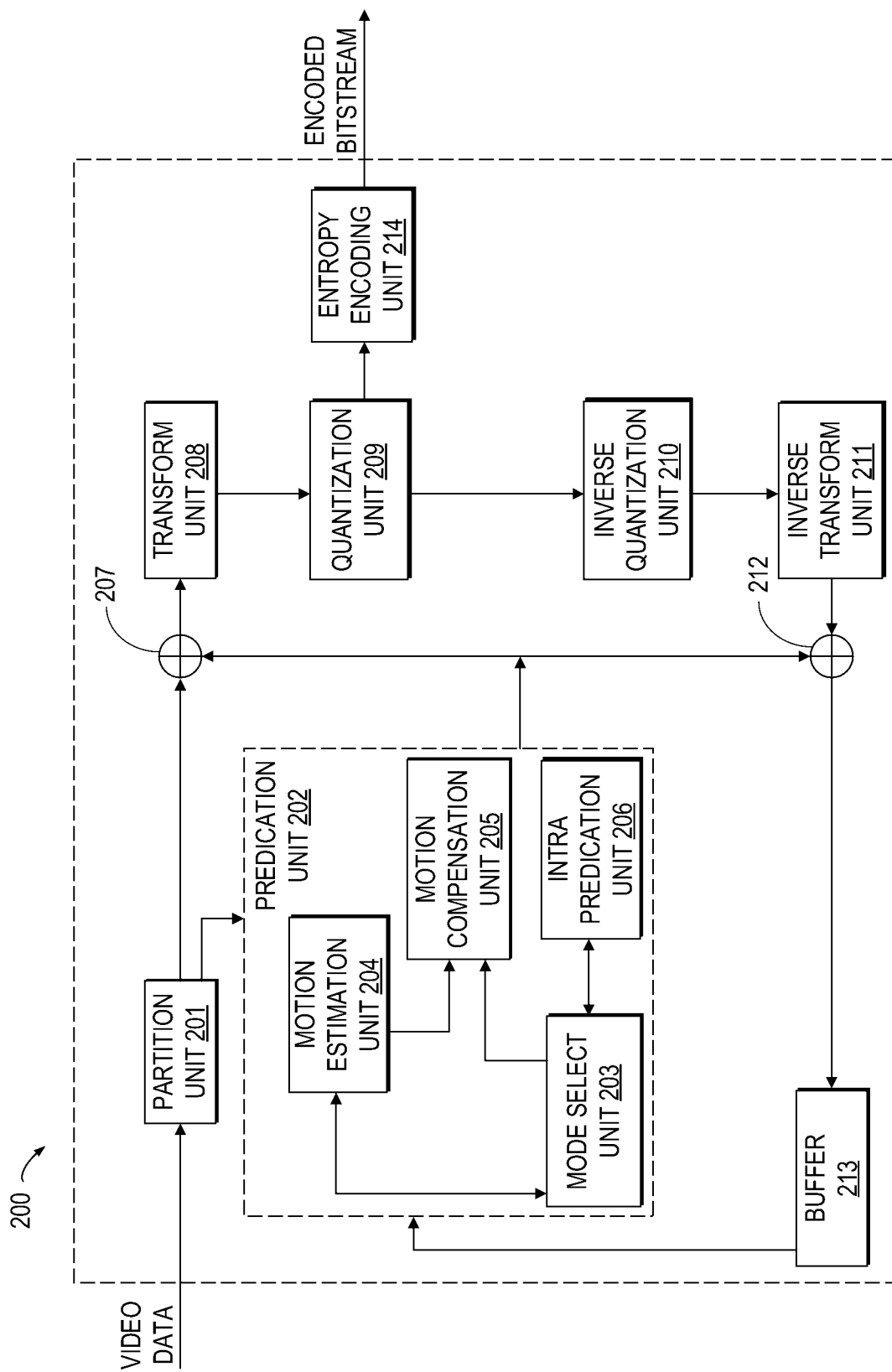


Fig. 2

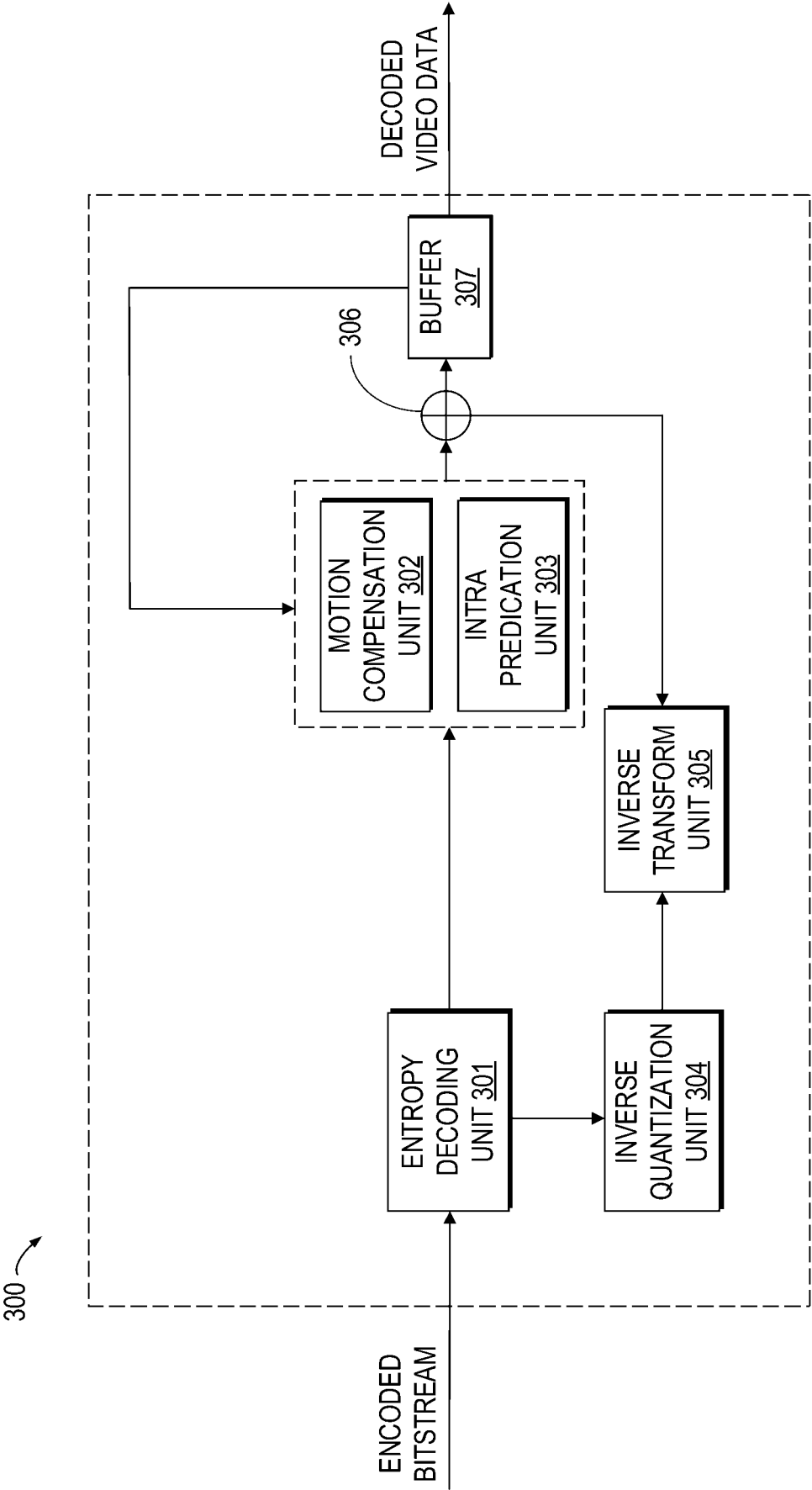


Fig. 3

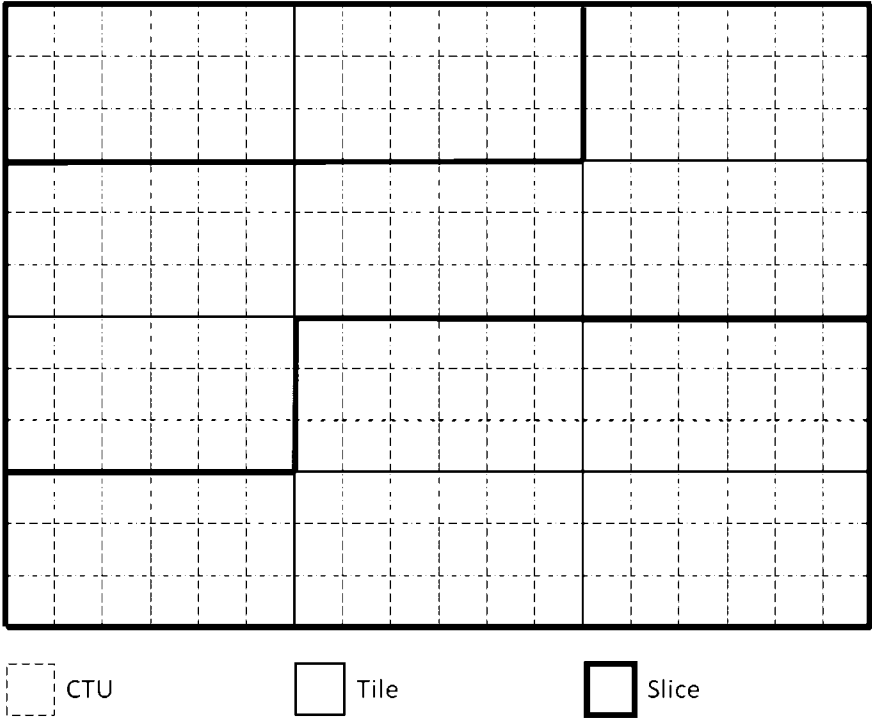


Fig. 4A

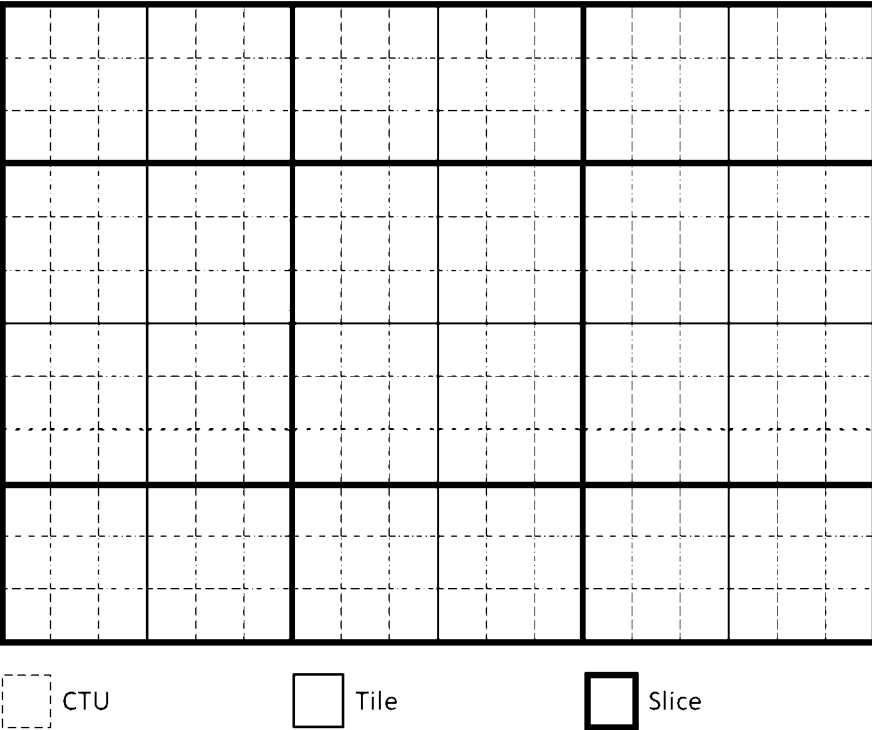
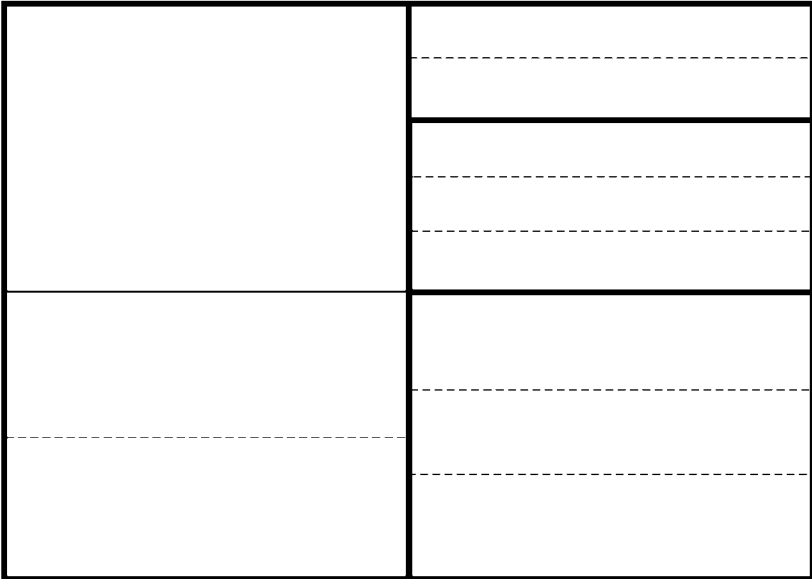


Fig. 4B



 Brick  Tile  Slice

Fig. 4C

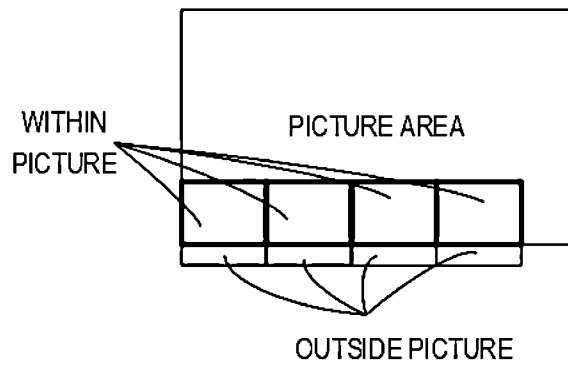


Fig. 4D

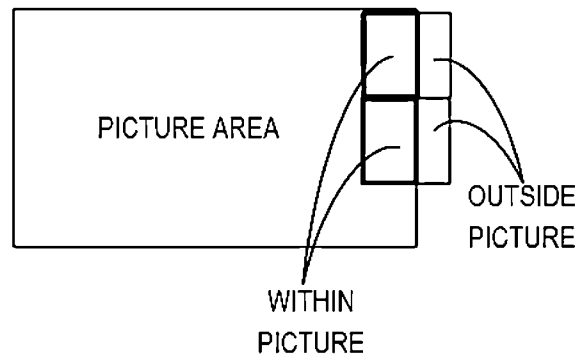


Fig. 4E

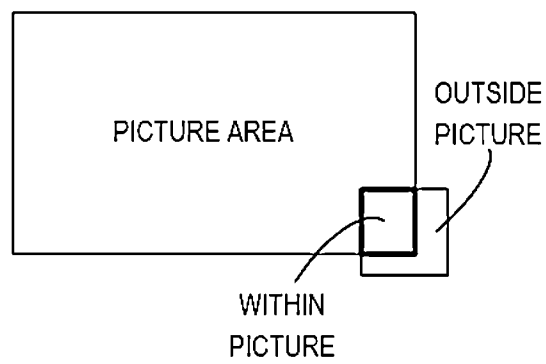


Fig. 4F

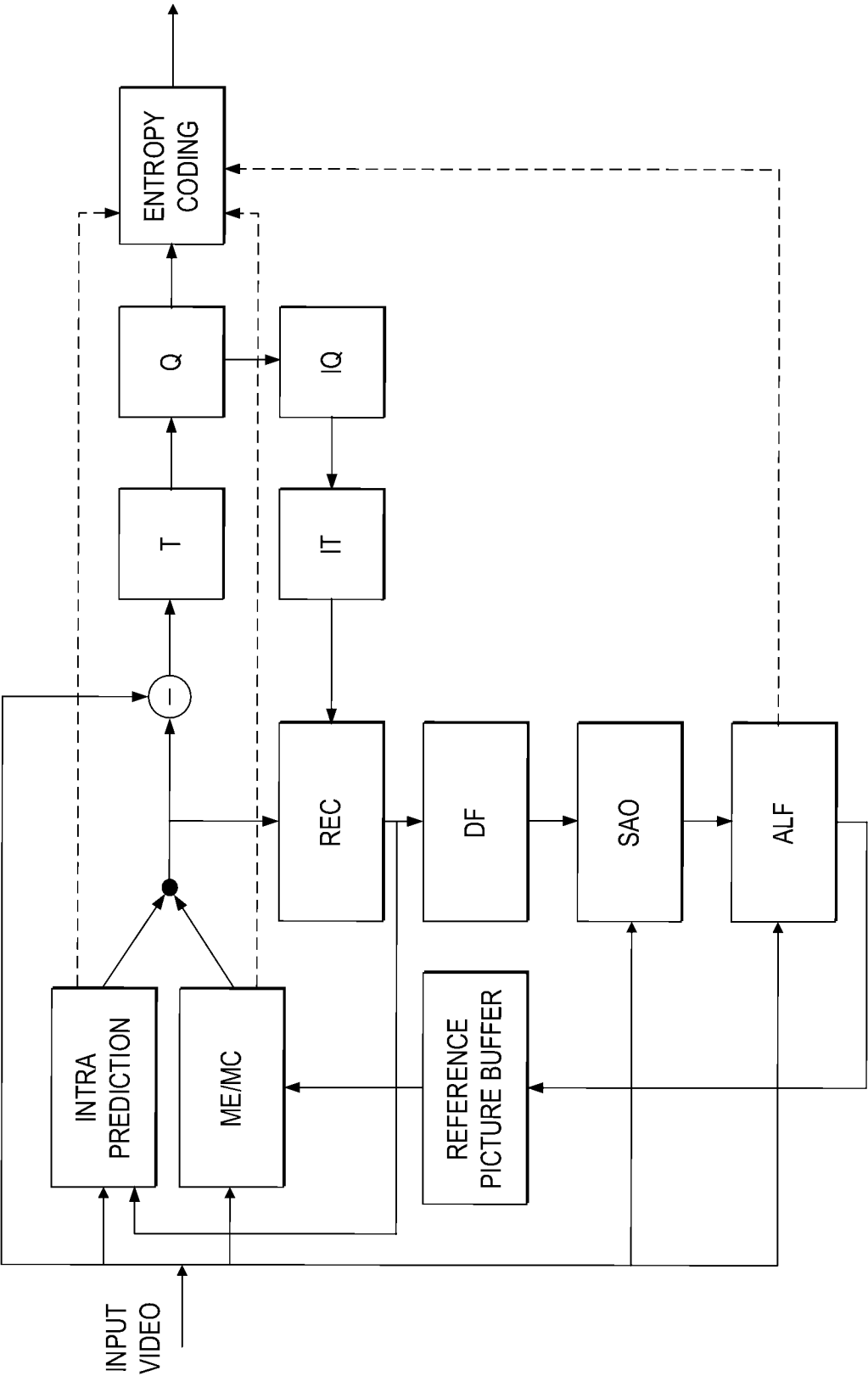


Fig. 5

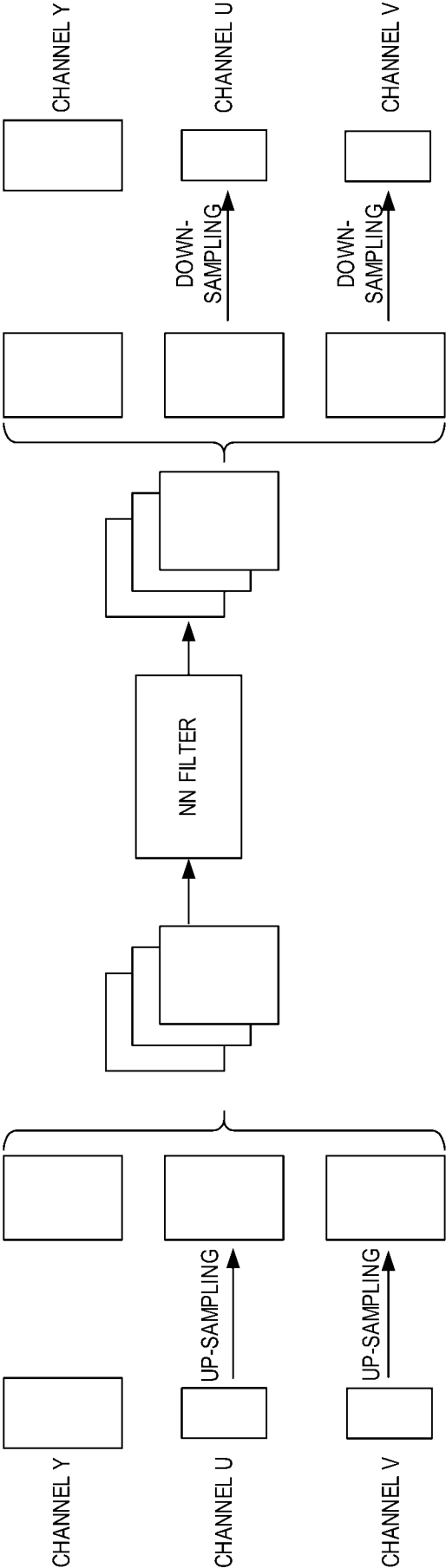


Fig. 6

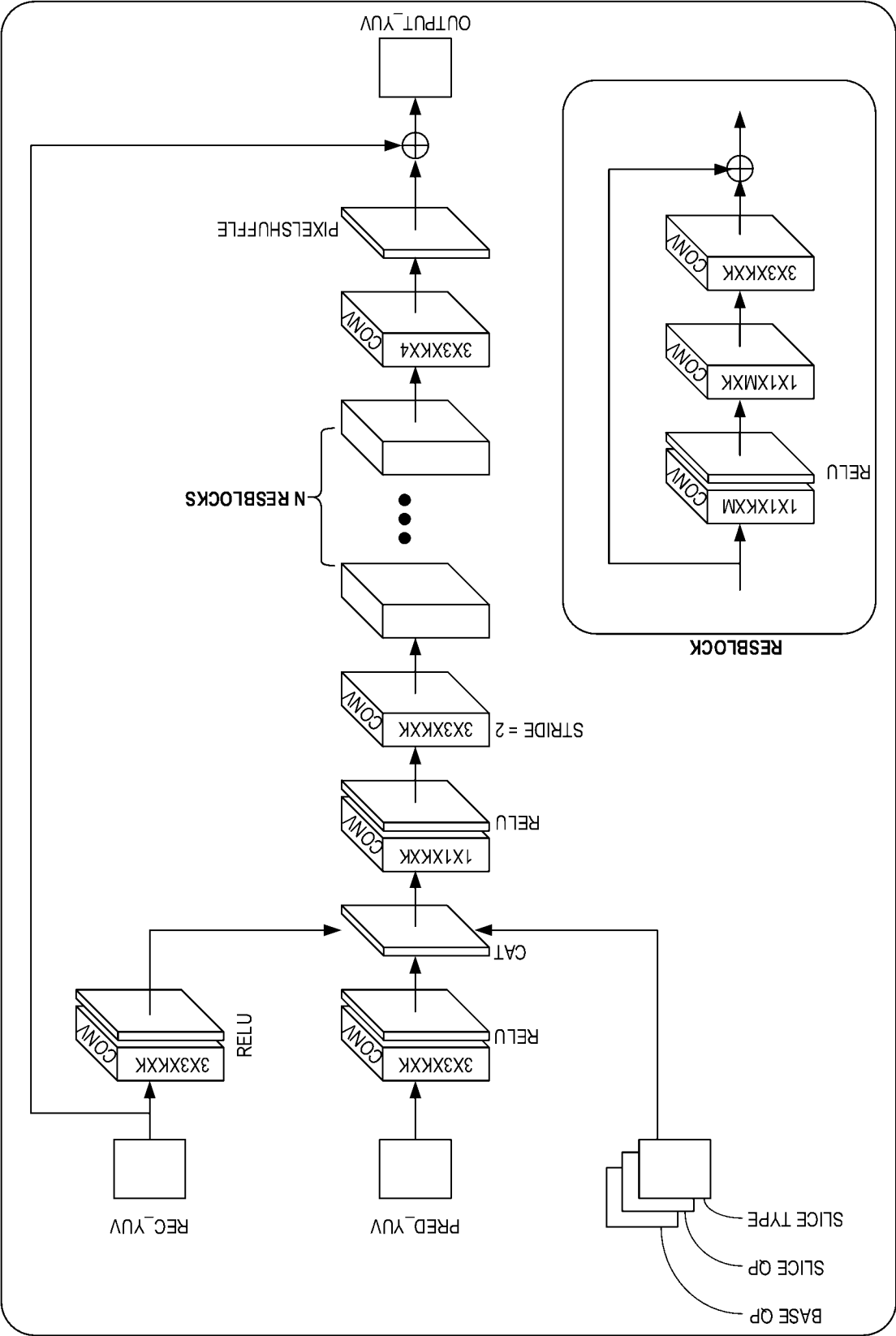


Fig. 7

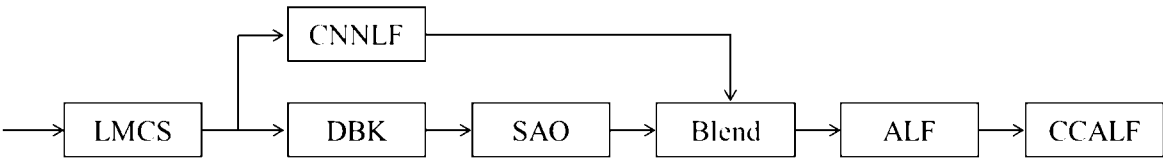


Fig. 8

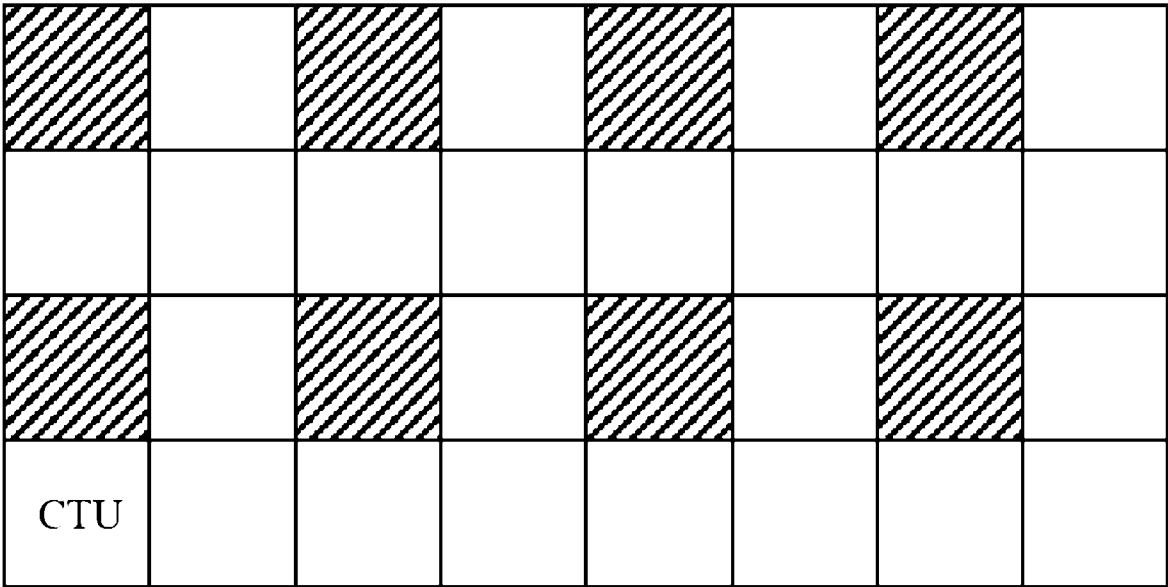


Fig. 9

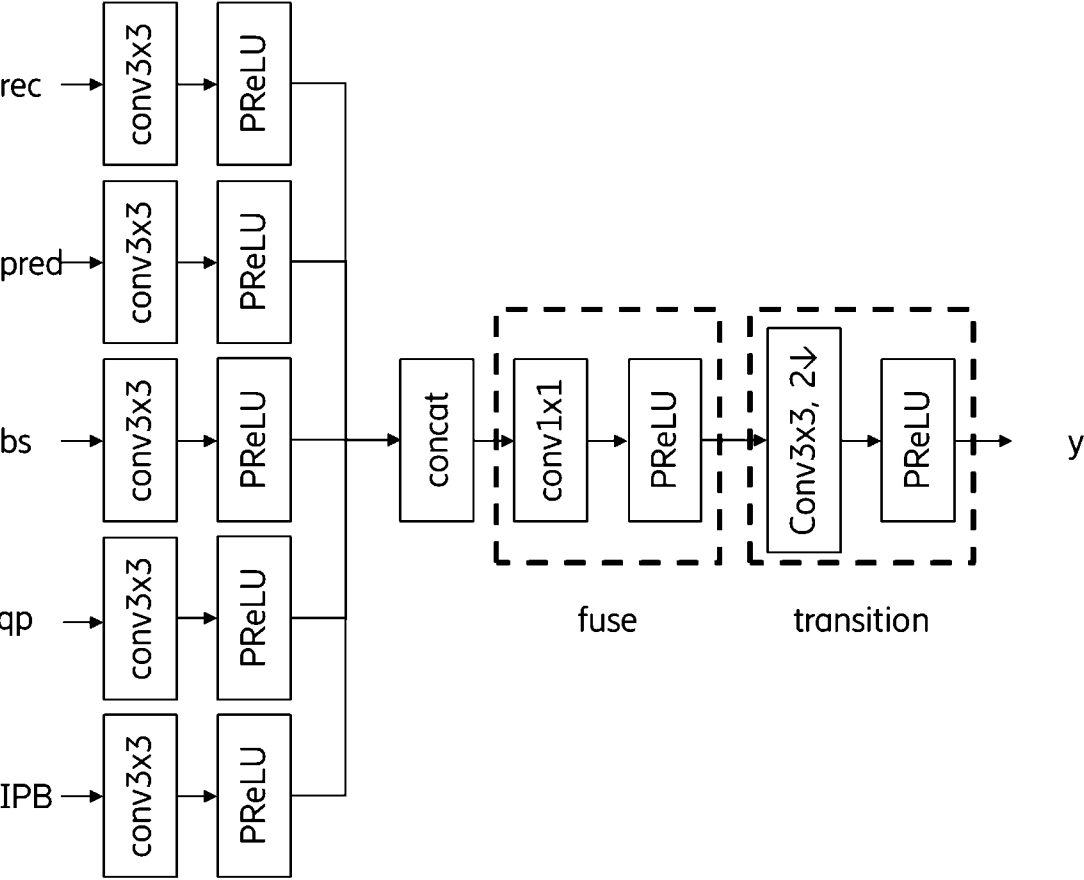


Fig. 10A

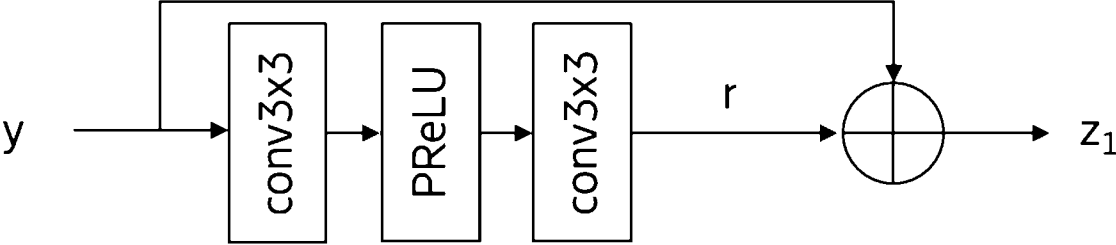


Fig. 10B

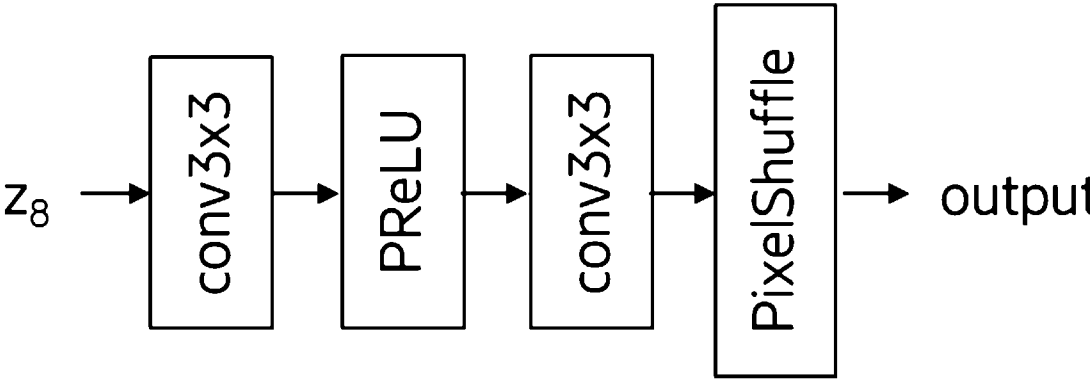


Fig. 10C

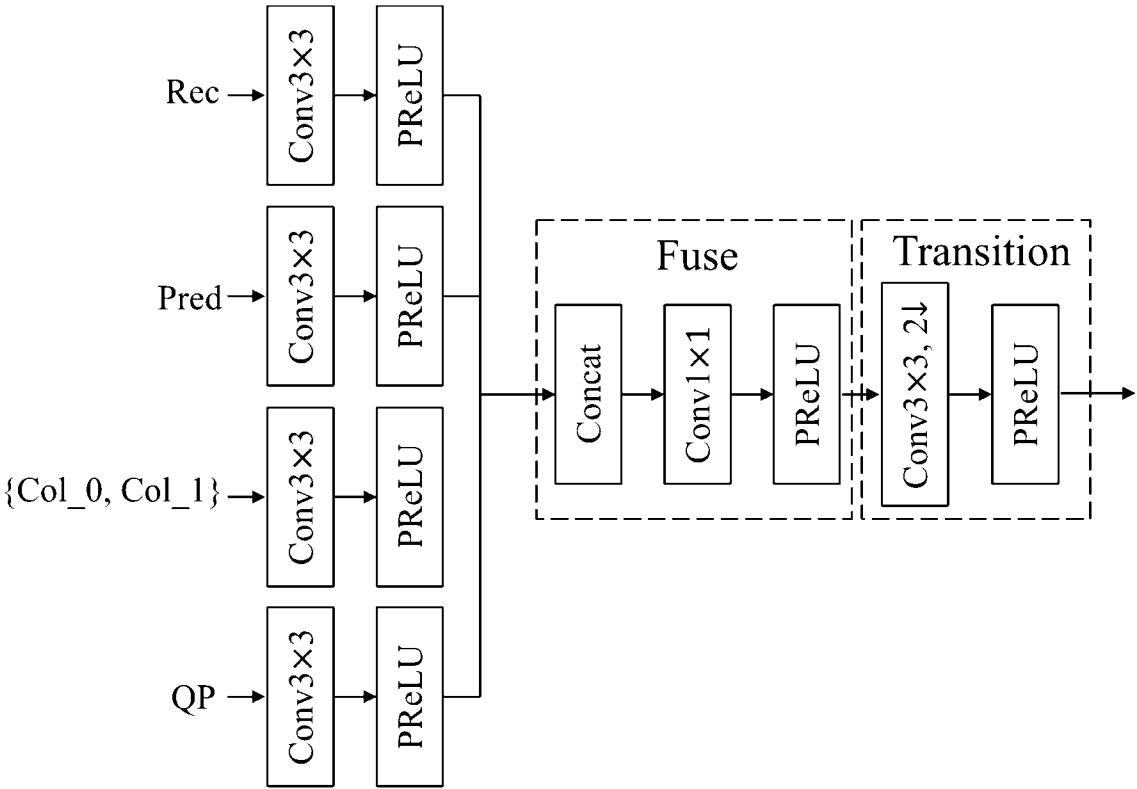


Fig. 11

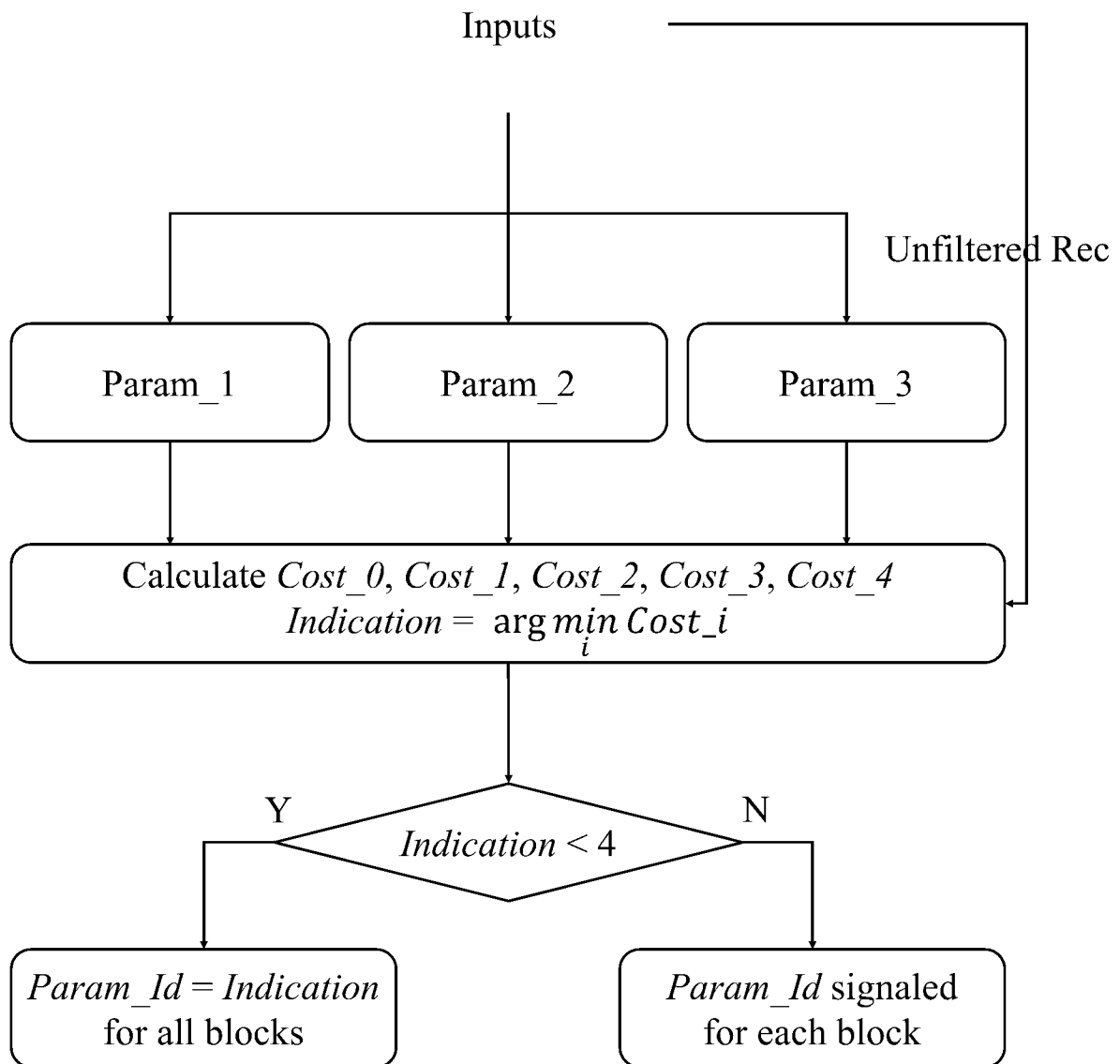


Fig. 12A

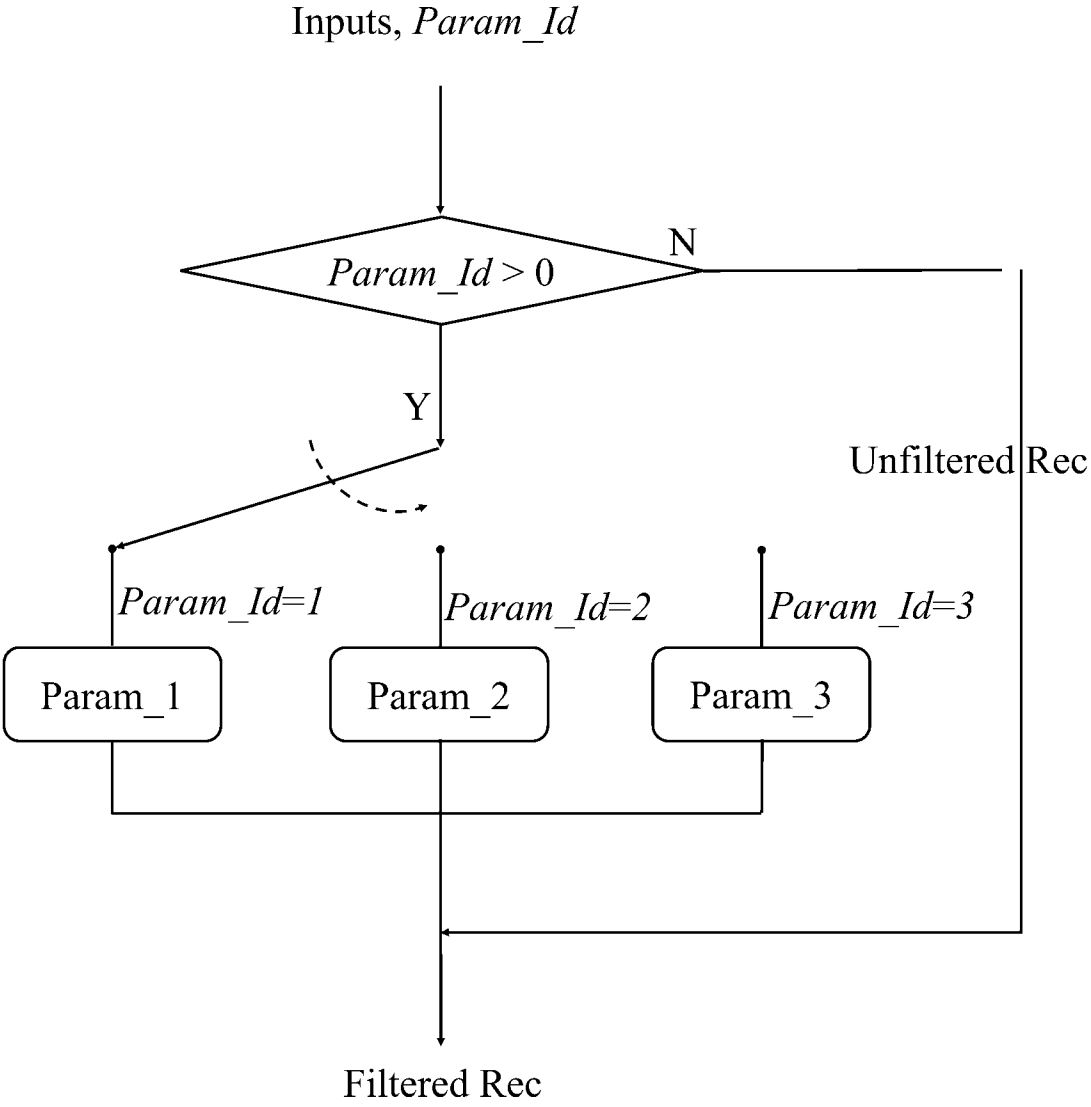


Fig. 12B

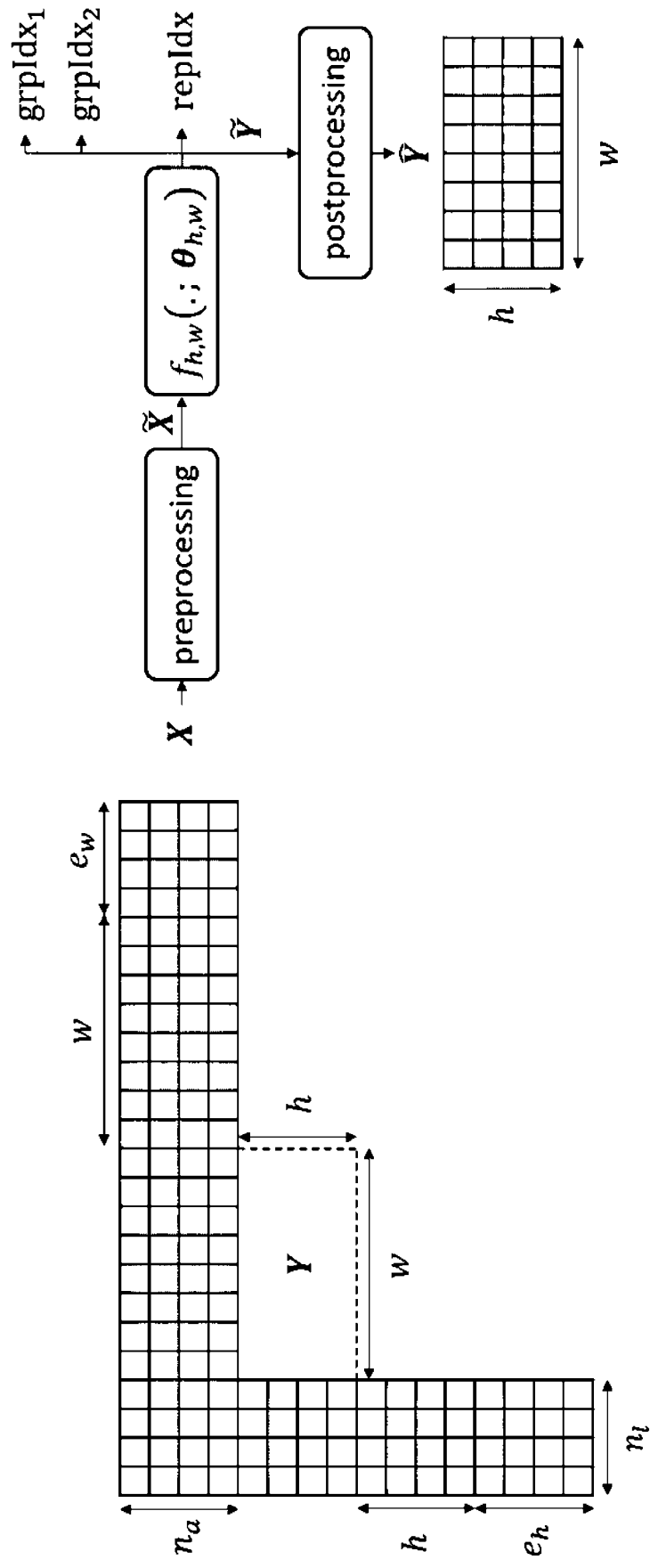


Fig. 13

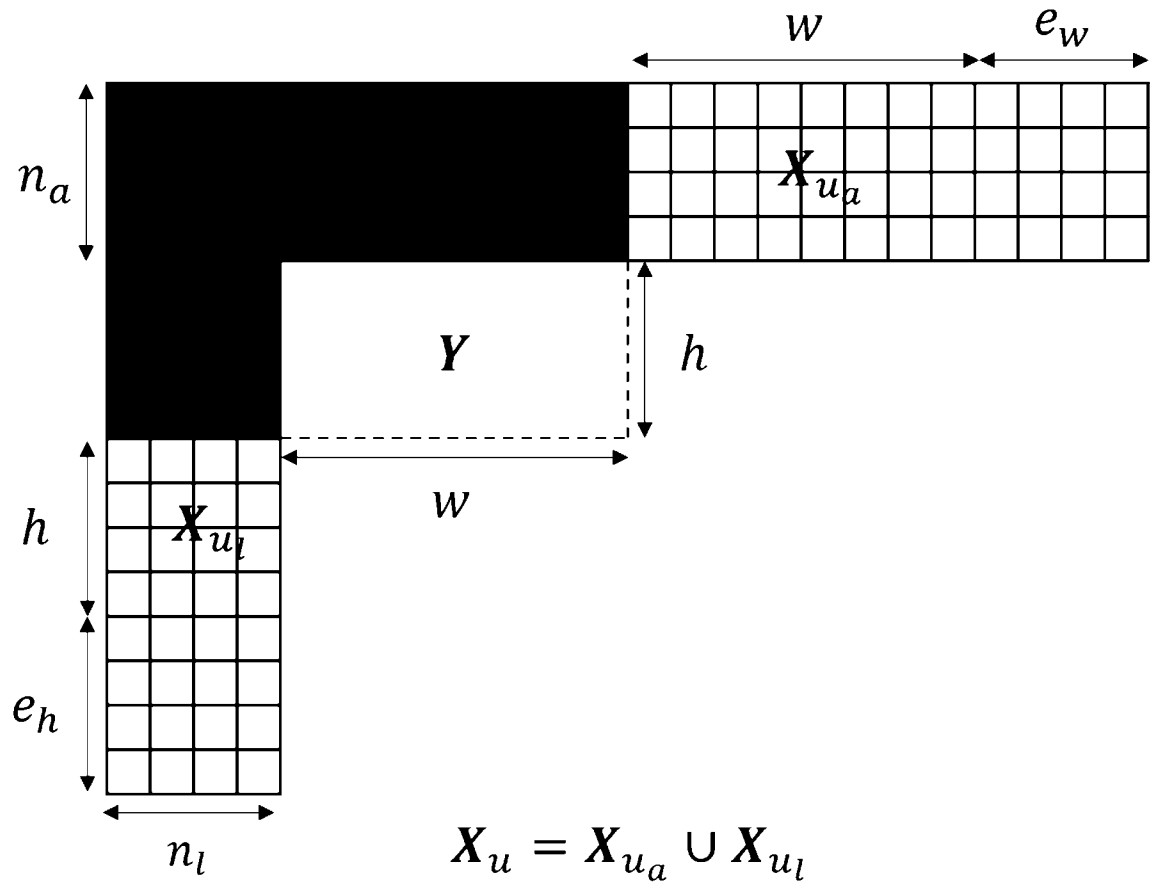


Fig. 14

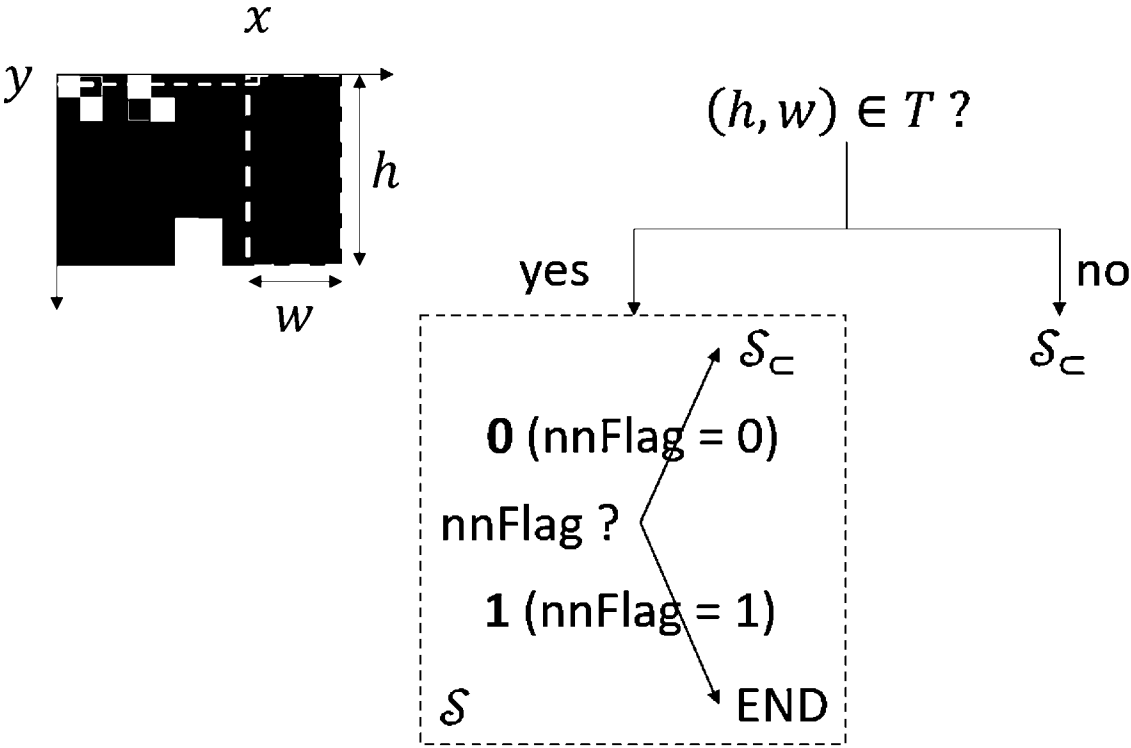


Fig. 15

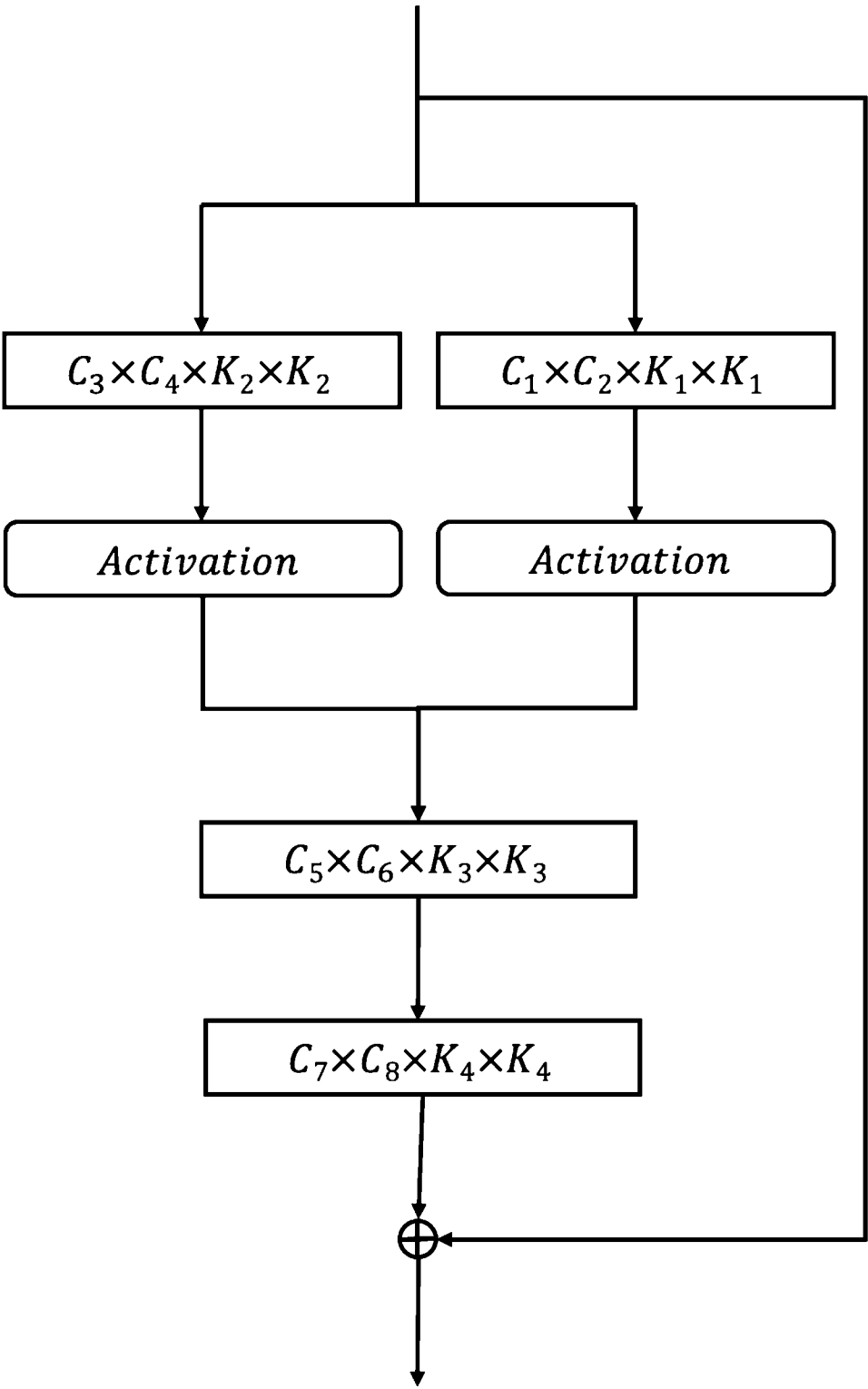


Fig. 16

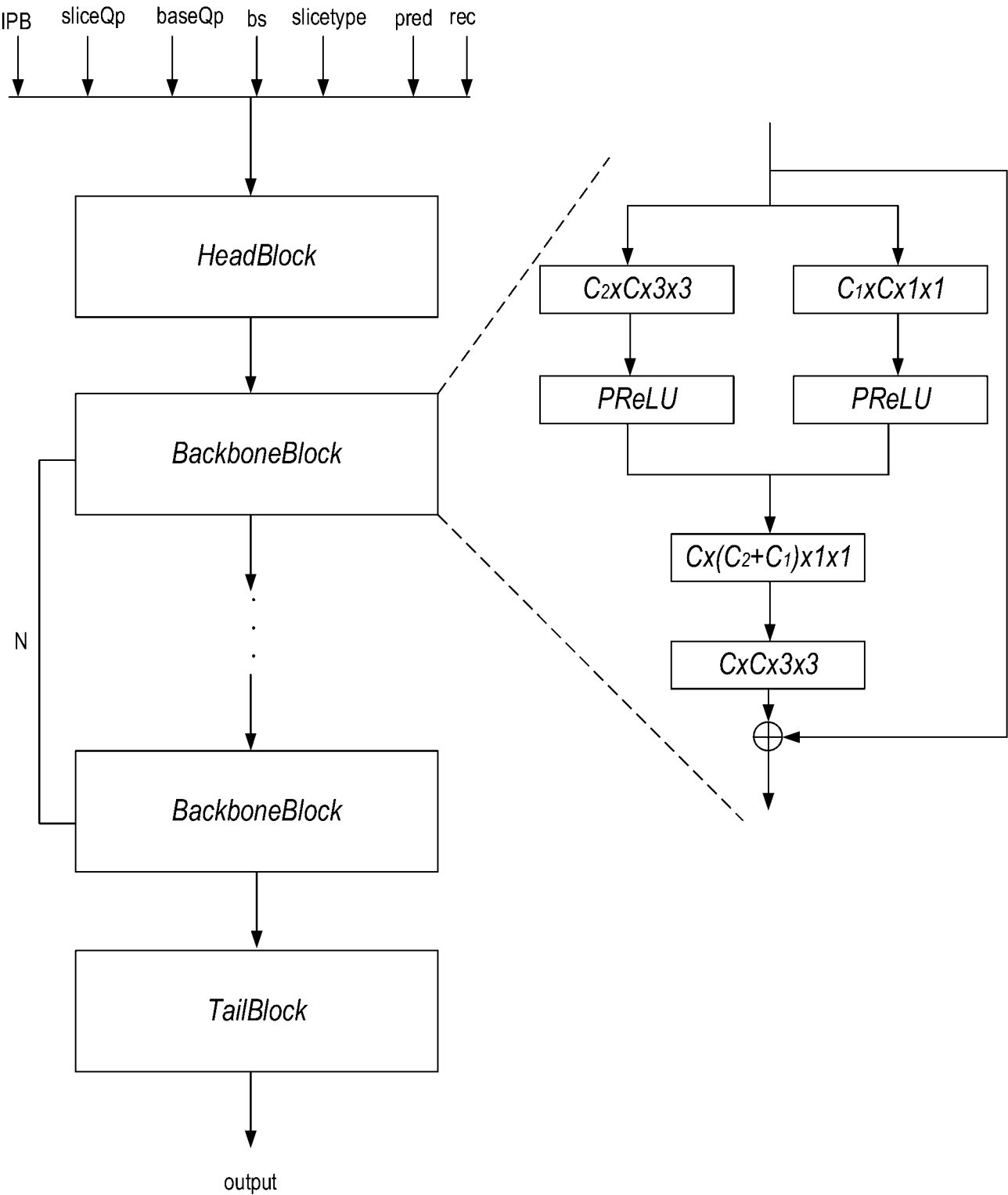


Fig. 17

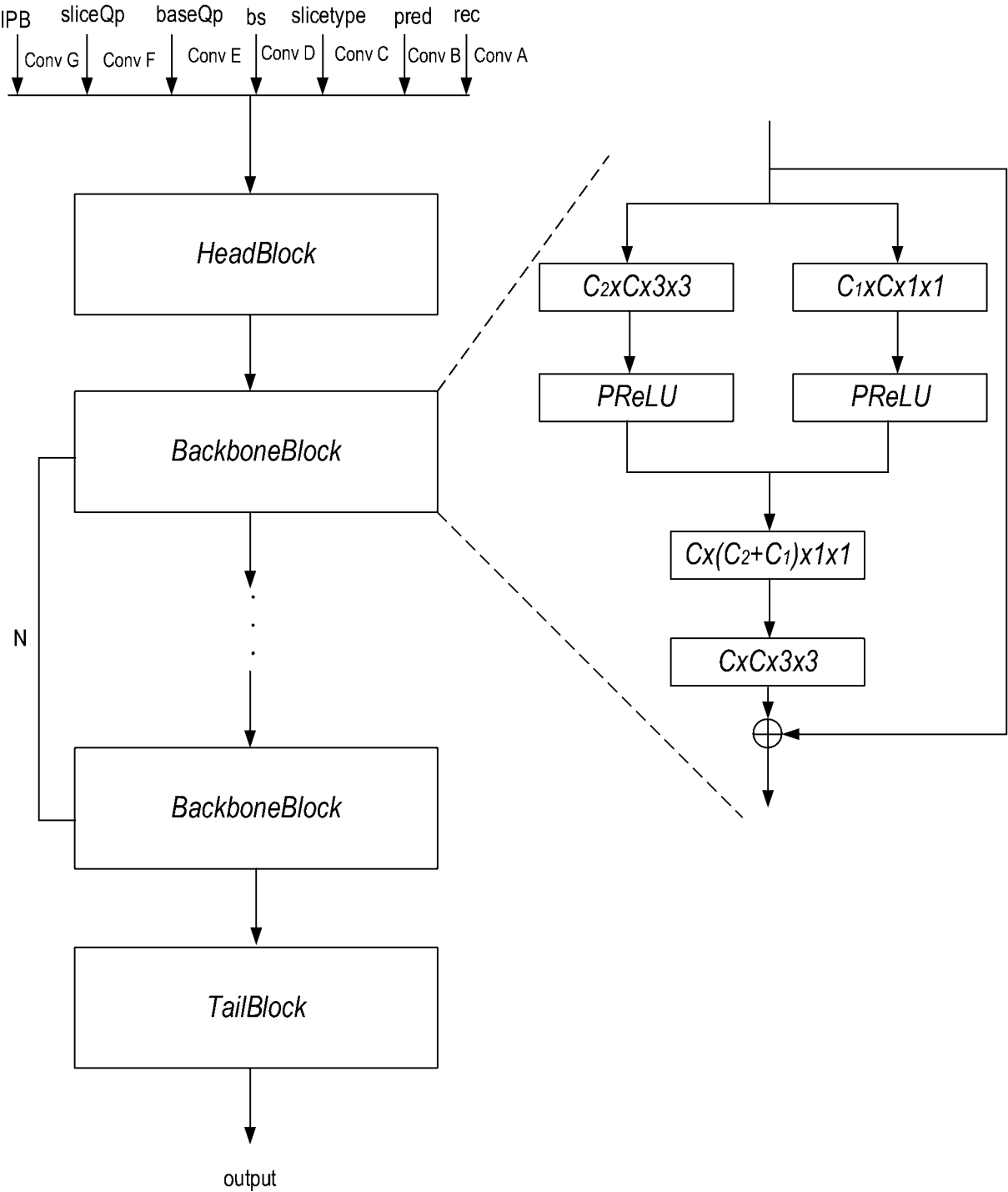


Fig. 18

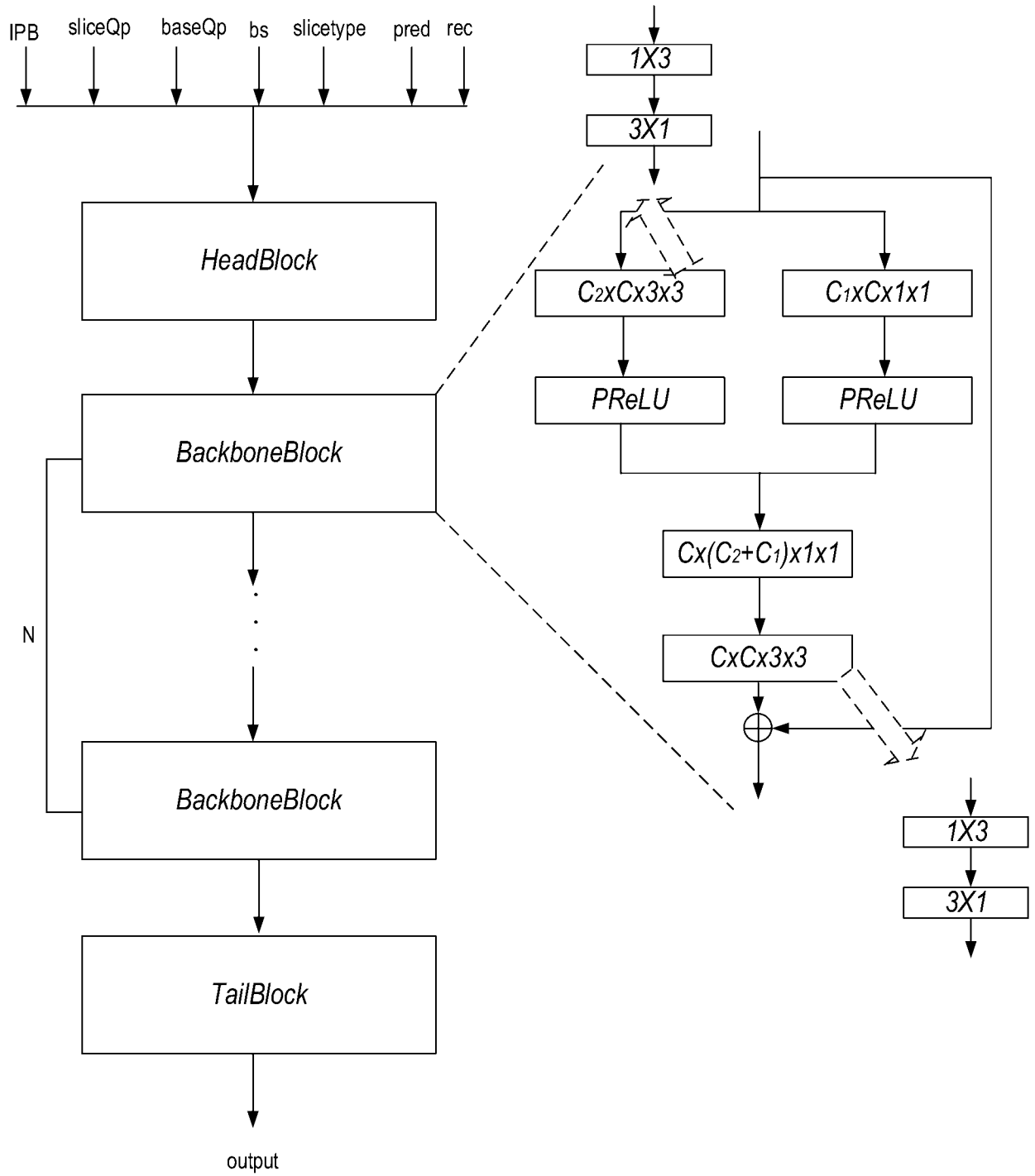


Fig. 19

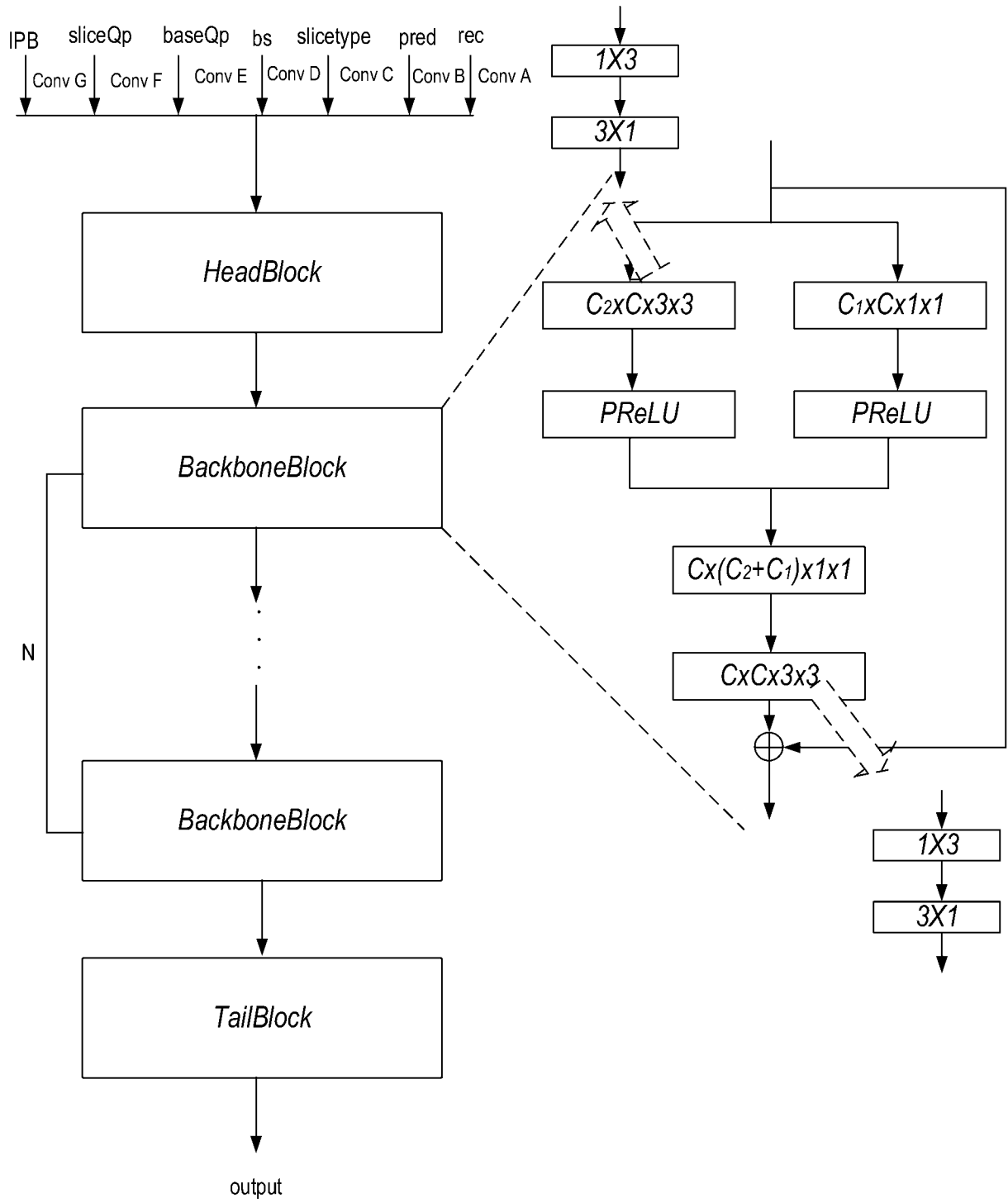


Fig. 20

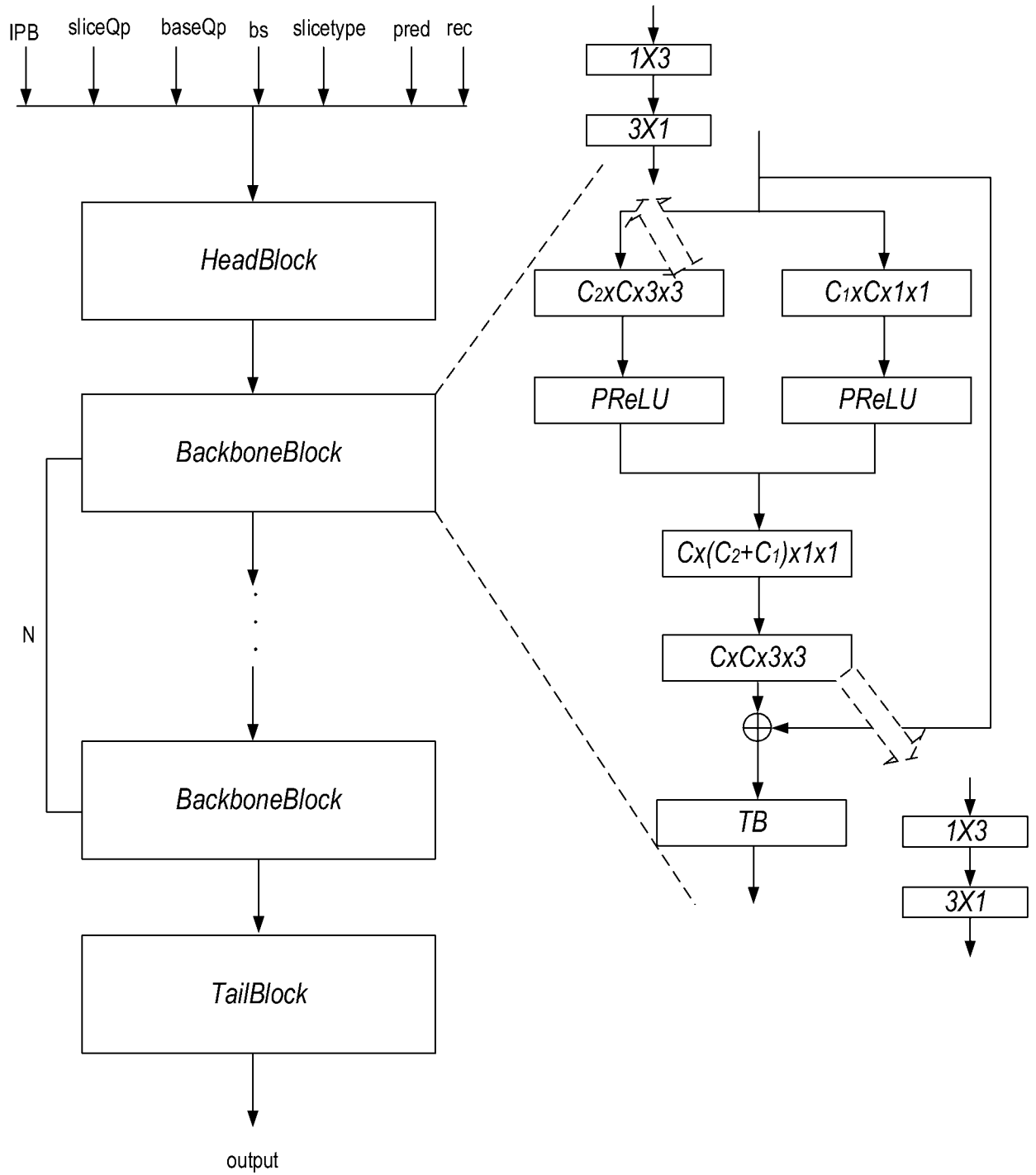


Fig. 21

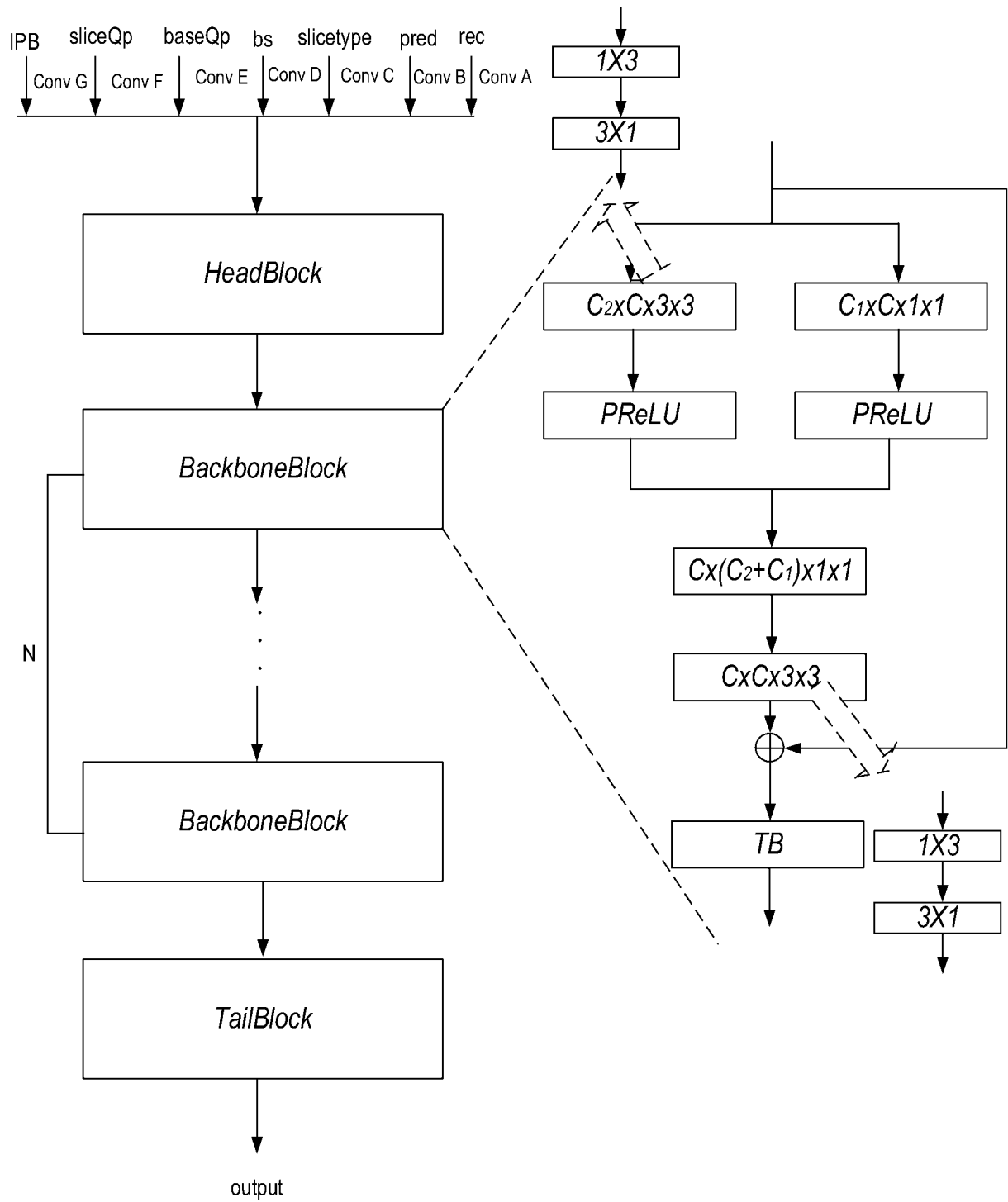


Fig. 22

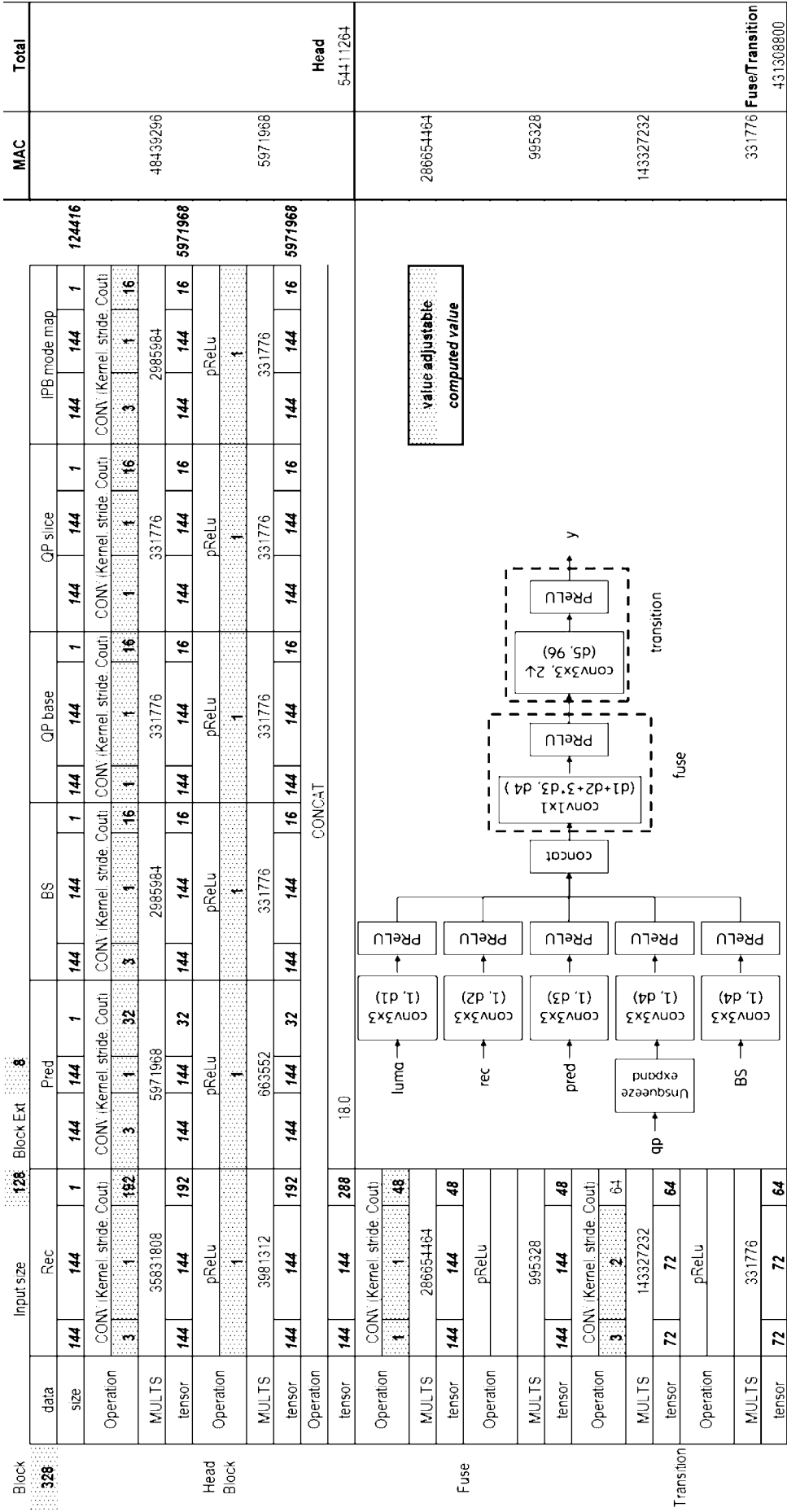


Fig. 23A

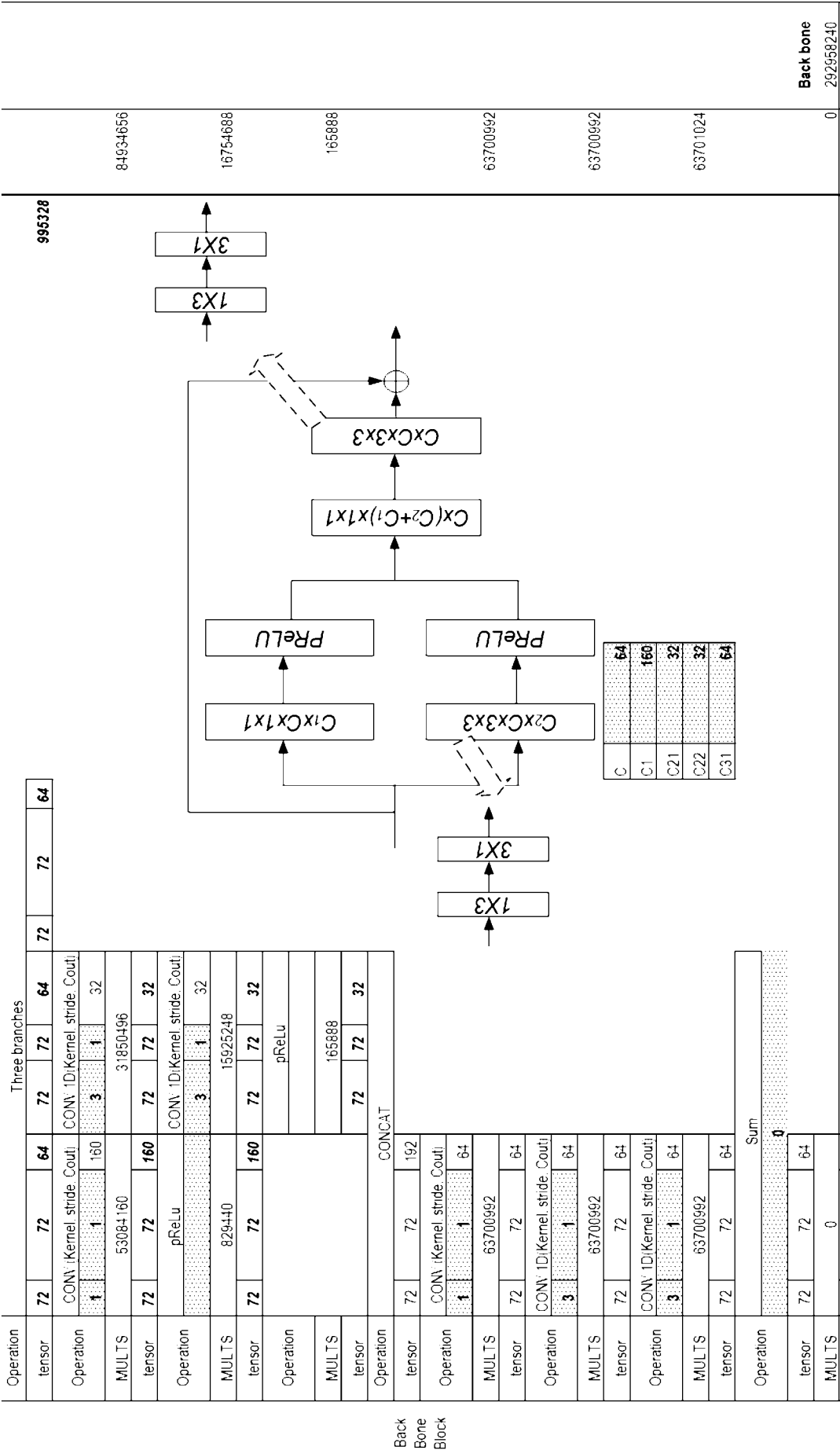
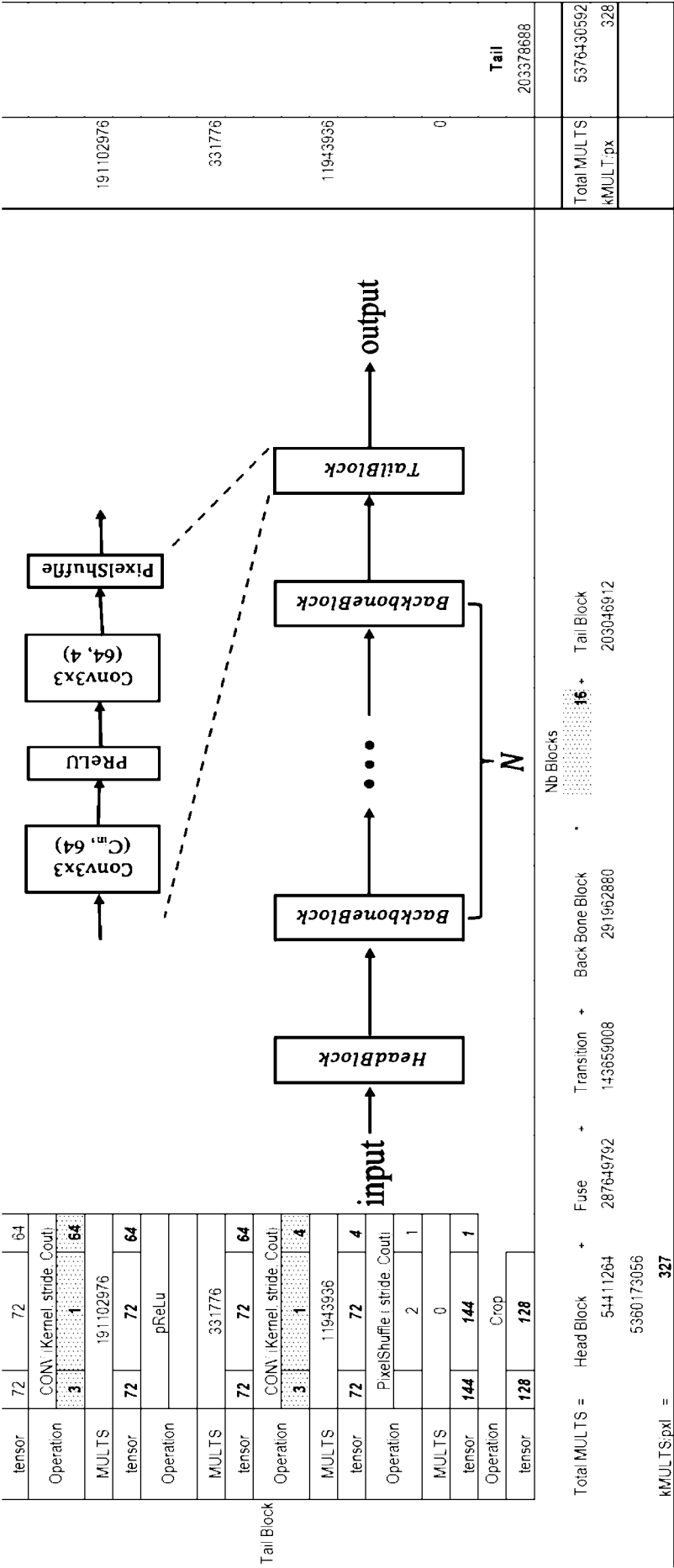


Fig. 23B



Total MULTS =

HeadBlock

+

Fuse

+

Transition

+

Back Bone Block

+

Tail Block

Nb Blocks

16

203048912

kMULTS:pxl =

327

Fig. 23C

Block 83	data size		72		2		72		72		2		72		72		3		72		72		1		72		72		1		144		144		1		67392	MAC	Total		
	Operation		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)		CONV (Kernel stride Cout)								
			3		1		192		3		1		32		3		1		16		1		1		16		1		1		16		3		2					16	
	MULTS		17915904																2239488								82944						746496								
	tensor		72		72		192		72		72		32		72		72		16		72		16		72		72		16		72		72		16						
	Operation				pReLU						pReLU								pReLU								pReLU						pReLU								
					1						1								1						1						1										
	MULTS		995328								165888								82944								82944						82944								
	tensor		72		72		192		72		72		32		72		72		16		72		16		72		72		16		72		72		16						
	Operation																		CONCAT																						
Fuse	tensor		72		288														18.0																		value adjustable computed value	71663616	Head 25546752		
	Operation		CONV (Kernel stride Cout)																																						
			1		1																																				
	MULTS																																								
	tensor		72		72		48																																		
	Operation																																								
	MULTS																																								
	tensor		72		72		48																																		
	Operation																																								
Transition	MULTS																																				35831808	248832	71663616		
	tensor		72		72		48																																		
	Operation		CONV (Kernel stride Cout)																																						
			3		2																																				
	MULTS																																								
	tensor		36		36		64																																		
	Operation																																								
	MULTS																																								
	tensor		36		36		64																																		

Fig. 24A

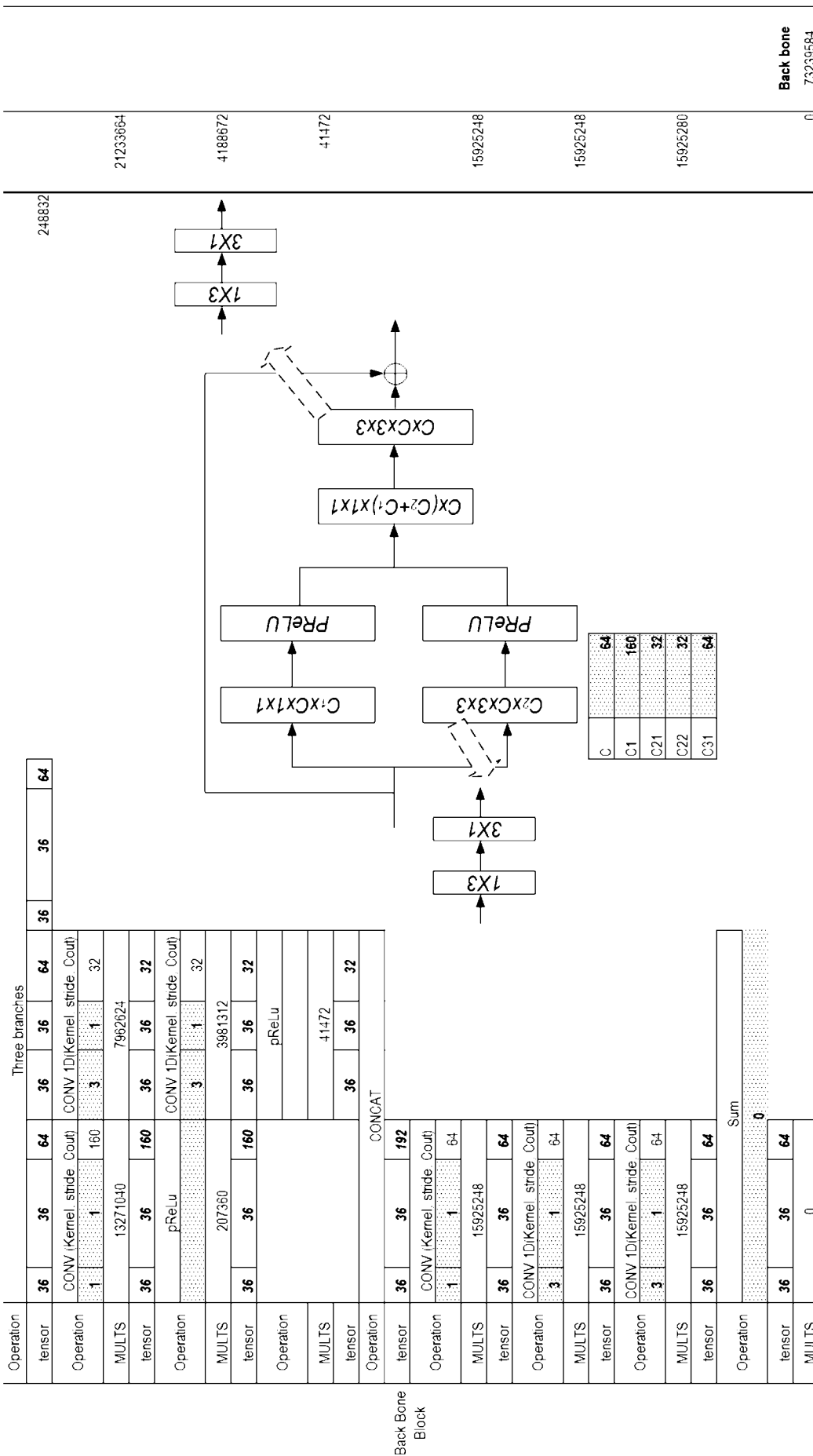


Fig. 24B

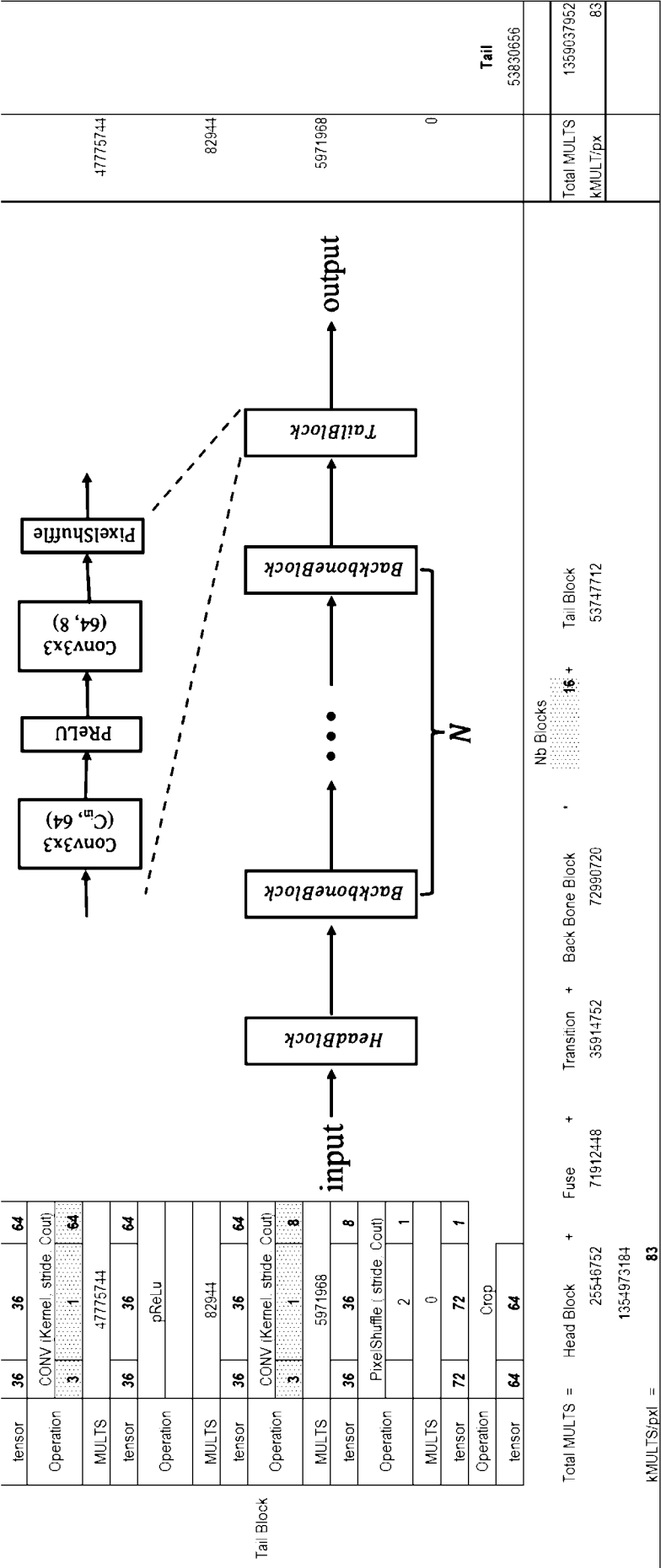


Fig. 24C

Block	Input size			128			Block Ext			8			MAC	Total	
	data size	Rec	Pred	BS	QP base	QP slice	IPB mode map								
335	Operation	3	1	192	3	1	32	3	1	16	1	16	3	1	16
	MULTS	107495424													
	tensor	144	144	192	144	144	32	144	144	16	144	16	144	144	1
	Operation	pReLU													
	Operation	1													
	MULTS	3981312													
	tensor	144	144	192	144	144	32	144	144	16	144	16	144	144	1
	Operation	CONCAT													
	tensor	144	144	288	18 0										
	Operation	CONV (Kernel stride Cout)													
Fuse	MULTS	286654464													
	tensor	144	144	48	value adjustable										
	Operation	pReLU													
	MULTS	995328													
	tensor	144	144	48	computed value										
	Operation	CONV (Kernel stride Cout)													
	MULTS	143327232													
	tensor	72	72	64	143327232										
	Operation	pReLU													
	MULTS	331776													
Transition	tensor	72	72	64	331776										
	Operation	Fuse/Transition													

Fig. 25A

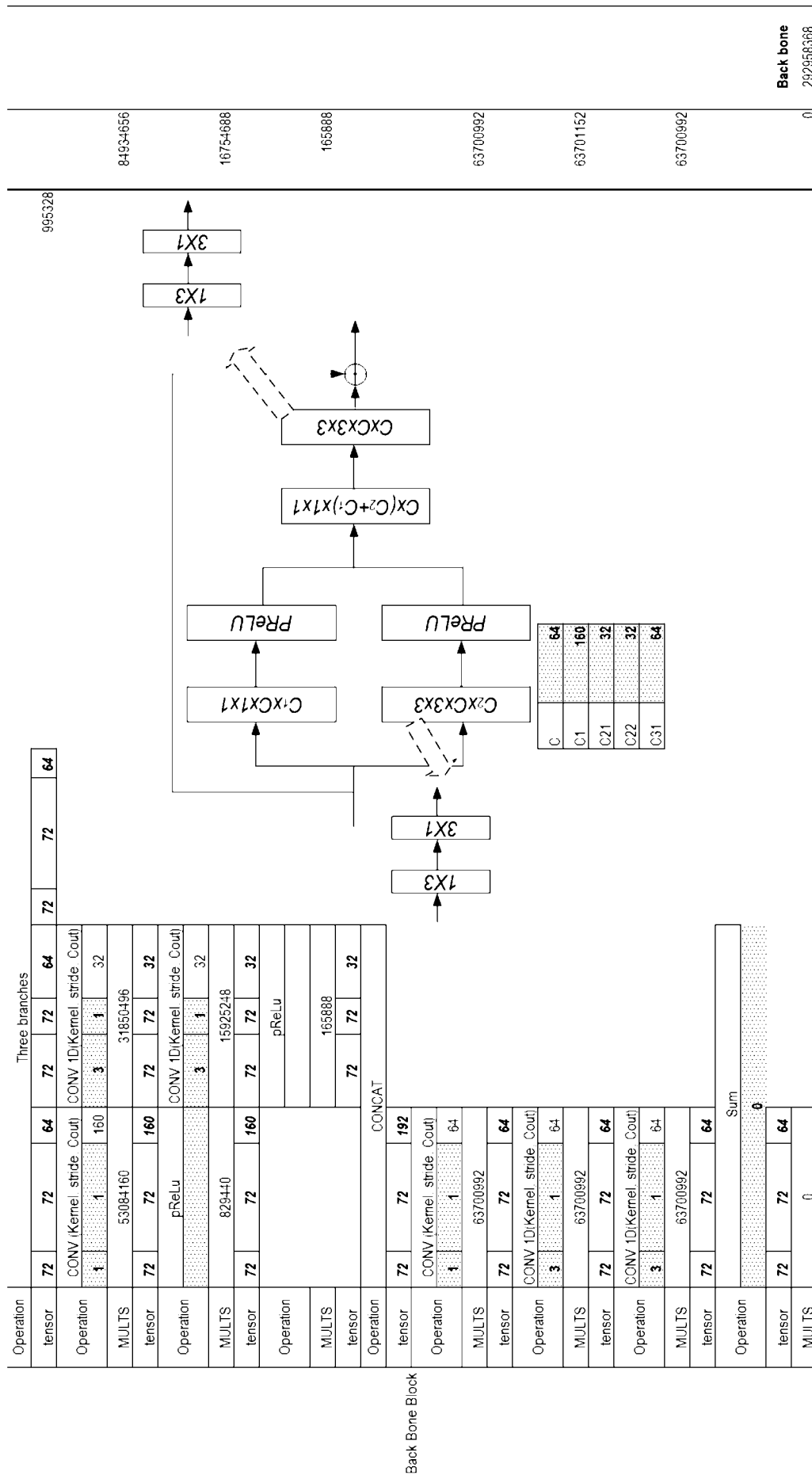


Fig. 25B

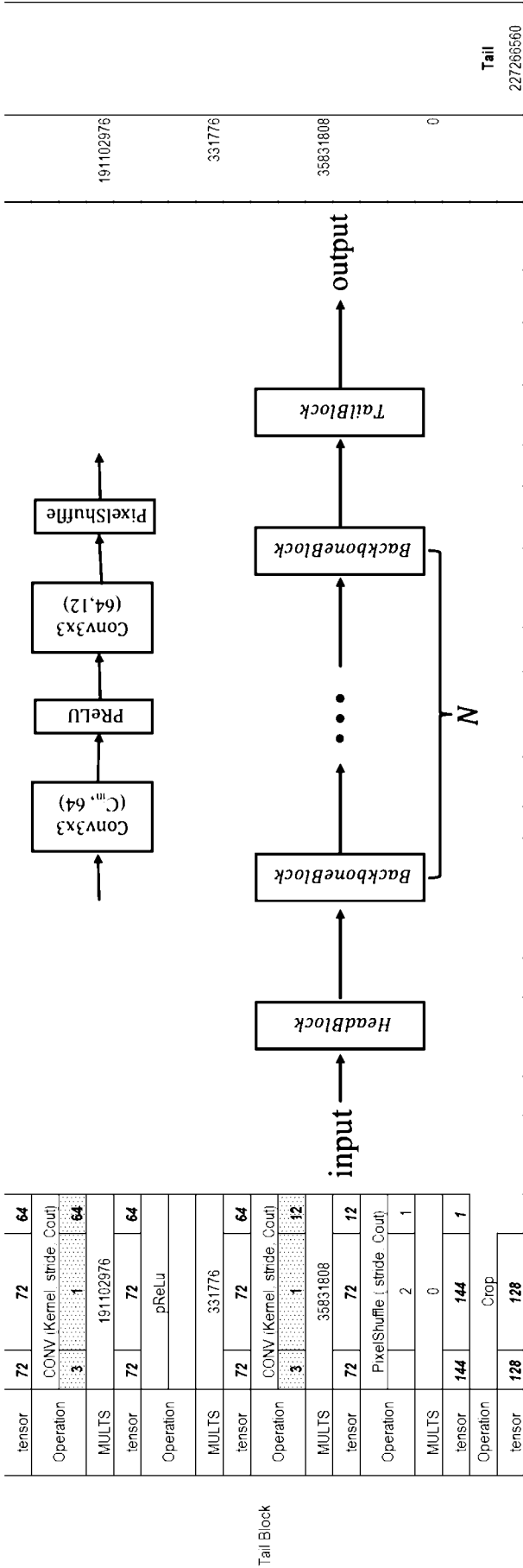
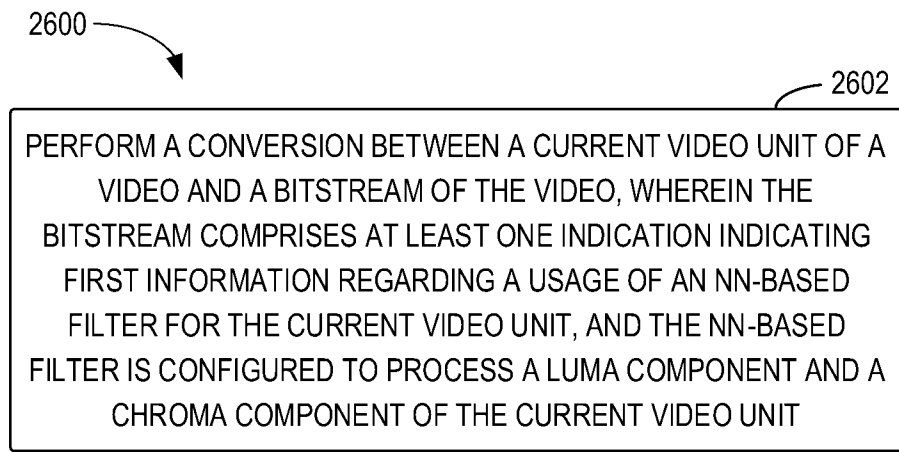


Fig. 25C

Tail Block

**Fig. 26**

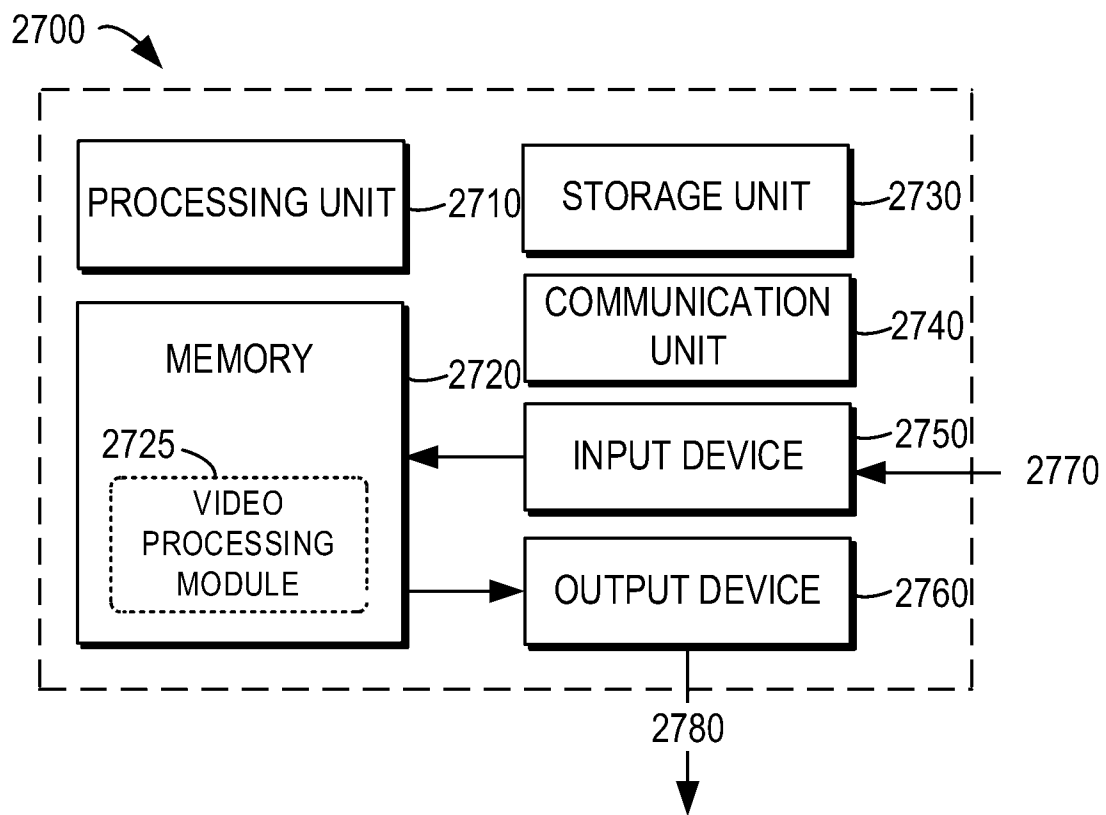


Fig. 27

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2024/095040

A. CLASSIFICATION OF SUBJECT MATTER H04N19/82(2014.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC: H04N Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) ENTXT, VEN, CNTXT, CNKI:video, bitstream, indication,neural network, NN, luma, chroma, CTU, CTB, QP, convolution		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2022329837 A1 (LEMON INC.) 13 October 2022 (2022-10-13) description, paragraphs [0004]-[0024], [0161]-[0415]	1-39, 86-94
X	US 2023051066 A1 (LEMON INC.) 16 February 2023 (2023-02-16) description, paragraphs [0004]-[0024], [0174]-[0435]	1-39, 86-94
A	WO 2023051653 A1 (BEIJING BYTEDANCE NETWORK TECH CO., LTD. et al.) 06 April 2023 (2023-04-06) the whole document	1-94
A	WO 2023056357 A1 (BYTEDANCE INC.) 06 April 2023 (2023-04-06) the whole document	1-94
A	US 2021329286 A1 (ALIBABA GROUP HOLDING LTD.) 21 October 2021 (2021-10-21) the whole document	1-94
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: “A” document defining the general state of the art which is not considered to be of particular relevance “D” document cited by the applicant in the international application “E” earlier application or patent but published on or after the international filing date “L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) “O” document referring to an oral disclosure, use, exhibition or other means “P” document published prior to the international filing date but later than the priority date claimed “T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art “&” document member of the same patent family		
Date of the actual completion of the international search 19 July 2024		Date of mailing of the international search report 01 August 2024
Name and mailing address of the ISA/CN CHINA NATIONAL INTELLECTUAL PROPERTY ADMINISTRATION 6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China		Authorized officer YUE,YongQing Telephone No. (+86) 010-62411575

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2024/095040

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2022329837	A1	13 October 2022	None			
US	2023051066	A1	16 February 2023	None			
WO	2023051653	A1	06 April 2023	None			
WO	2023056357	A1	06 April 2023	None			
US	2021329286	A1	21 October 2021	US	2024146951	A1	02 May 2024
				WO	2021211966	A1	21 October 2021
				US	11902561	B2	13 February 2024