(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
02 February 2023 (02.02.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/009747 A1

(51) International Patent Classification:
G06T 9/00 (2006.01) *G06N 3/04* (2006.01)
H04N 19/117 (2014.01) *G06N 3/08* (2006.01)

(21) International Application Number:
PCT/US2022/038711

(22) International Filing Date:
28 July 2022 (28.07.2022)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/227,314 29 July 2021 (29.07.2021) US

(71) Applicant: **BEIJING DAJIA INTERNET INFORMATION TECHNOLOGY CO., LTD.** [CN/CN]; Room 101D1-7, 1st Floor, Building 1, No. 6, Shangdi West Road, Haidian District, Beijing, Beijing 100085 (CN).

(72) Inventor; and

(71) Applicant *(for US only)*: **CHEN, Wei** [CN/US]; 12569 Carmel Canyon Road, San Diego, California 92130 (US).

(72) Inventors: **XIU, Xiaoyu**; 6691 Aliso Valley Way, San Diego, California 92130 (US). **CHEN, Yi-Wen**; 8910 University Center Lane, Suite 400, San Diego, California 92122 (US). **JHU, Hong-Jheng**; 8910 University Center Lane, Suite 400, San Diego, California 92122 (US). **KUO, Che-Wei**; Room 101D1-7, 1st Floor, Building 1, No. 6, Shangdi West Road, Haidian District, Beijing, 100085 (CN). **WANG, Xianglin**; 8910 University Center Lane, Suite 400, San Diego, California 92122 (US). **YU, Bing**; Room 101D1-7, 1st Floor, Building 1, No. 6, Shangdi West Road, Beijing, 100085 (CN).

(74) Agent: **HAO, Tan**; Arch & Lake LLP, 203 N. LaSalle Street, Suite 2100, Chicago, Illinois 60601 (US).

(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,

(54) Title: NETWORK BASED IMAGE FILTERING FOR VIDEO CODING

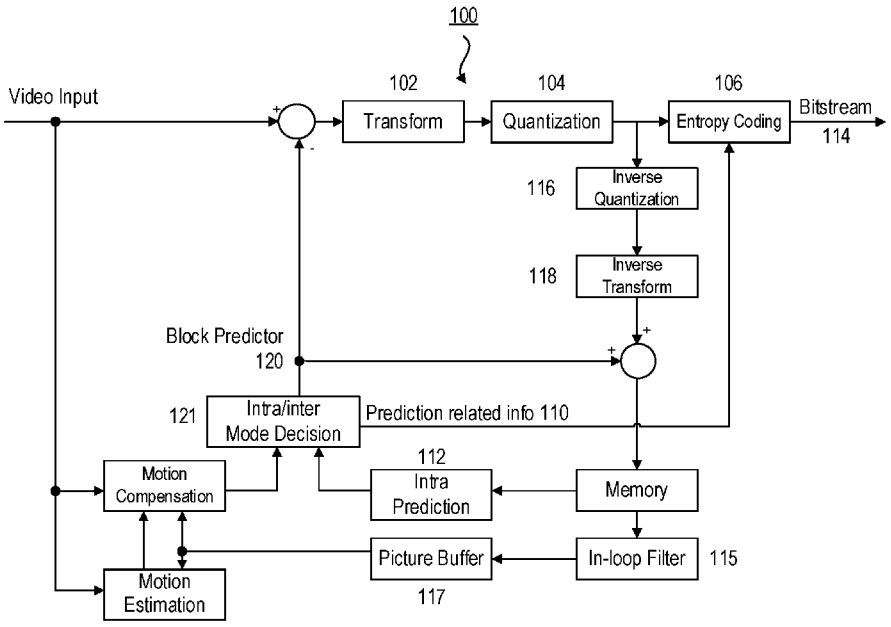


FIG. 1

(57) Abstract: A method and an apparatus for hybrid training of neural networks for video coding are provided. The method includes: obtaining, in an offline training stage, an offline trained network by training a neural network offline; and refining, in an online training stage, a plurality of neural network layers with constraint on a plurality of parameters of the plurality of neural network layers, where the plurality of neural network layers may include at least one neural network layer in the offline trained network or in a simple neural network connected to the offline trained network.

WO 2023/009747 A1

[Continued on next page]

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, ME, MG, MK, MN, 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, GH, GM, KE, LR, LS, MW, MZ, NA, RW, 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, 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).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

NETWORK BASED IMAGE FILTERING FOR VIDEO CODING

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims priority to U.S. Provisional Application No. 63/227,314, entitled "Methods and Apparatus for Hybrid Training of Neural Networks for Video Coding," filed on July 29, 2021, the entirety of which is incorporated by reference for all purposes.

FIELD

[0002] The present disclosure relates to video coding, and in particular but not limited to, methods and apparatus for hybrid training of neural networks for video coding.

BACKGROUND

[0003] Various video coding techniques may be used to compress video data. Video coding is performed according to one or more video coding standards. For example, video coding standards include versatile video coding (VVC), joint exploration test model (JEM), high-efficiency video coding (H.265/HEVC), advanced video coding (H.264/AVC), moving picture expert group (MPEG) coding, or the like. Video coding generally utilizes prediction methods (e.g., inter-prediction, intra-prediction, or the like) that take advantage of redundancy present in video images or sequences. An important goal of video coding techniques is to compress video data into a form that uses a lower bit rate, while avoiding or minimizing degradations to video quality.

[0004] The first version of the HEVC standard was finalized in October 2013, which offers approximately 50% bit-rate saving or equivalent perceptual quality compared to the prior generation video coding standard H.264/MPEG AVC. Although the HEVC standard provides significant coding improvements than its predecessor, there is evidence that superior coding efficiency can be achieved with additional coding tools over HEVC. Based on that, both VCEG and MPEG started the exploration work of new coding technologies for future video coding standardization. one Joint Video Exploration Team (JVET) was formed in Oct. 2015 by ITU-T VECG and ISO/IEC MPEG to begin significant study of advanced technologies that could enable substantial enhancement of coding efficiency. One reference software called joint

exploration model (JEM) was maintained by the JVET by integrating several additional coding tools on top of the HEVC test model (HM).

[0005] The joint call for proposals (CfP) on video compression with capability beyond HEVC was issued by ITU-T and ISO/IEC. 23 CfP responses were received and evaluated at the 10-th JVET meeting, which demonstrated compression efficiency gain over the HEVC around 40%. Based on such evaluation results, the JVET launched a new project to develop the new generation video coding standard that is named as Versatile Video Coding (VVC). One reference software codebase, called VVC test model (VTM), was established for demonstrating a reference implementation of the VVC standard.

SUMMARY

[0006] The present disclosure provides examples of techniques relating to improving video coding efficiency using a hybrid training on neural networks for video coding.

[0007] According to a first aspect of the present disclosure, there is provided a method for hybrid training of neural networks for video coding. The method includes: obtaining, in an offline training stage, an offline trained network by training a neural network offline; and refining, in an online training stage, a plurality of neural network layers with constraint on a plurality of parameters of the plurality of neural network layers, wherein the plurality of neural network layers comprise at least one neural network layer in the offline trained network or in a simple neural network connected to the offline trained network.

[0008] According to a second aspect of the present disclosure, there is provided an apparatus for hybrid training of neural networks for video coding. The apparatus includes one or more processors and a memory configured to store instructions executable by the one or more processors. Further, the one or more processors, upon execution of the instructions, are configured to perform the method according to the first aspect.

[0009] According to a third aspect of the present disclosure, there is provided a non-transitory computer-readable storage medium storing computer-executable instructions that, when executed by one or more computer processors, causing the one or more computer processors to perform the method according to the first aspect.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] A more particular description of the examples of the present disclosure will be rendered by reference to specific examples illustrated in the appended drawings. Given that these drawings depict only some examples and are not therefore considered to be limiting in scope, the examples will be described and explained with additional specificity and details through the use of the accompanying drawings.

[0011] FIG. 1 is a block diagram illustrating a block-based video encoder in accordance with some implementations of the present disclosure.

[0012] FIG. 2 is a block diagram illustrating a block-based video decoder in accordance with some implementations of the present disclosure.

[0013] FIG. 3A is schematic diagram illustrating quaternary partitioning tree splitting mode in accordance with some implementations of the present disclosure.

[0014] FIG. 3B is schematic diagram illustrating vertical binary partitioning tree splitting mode in accordance with some implementations of the present disclosure.

[0015] FIG. 3C is schematic diagram illustrating horizontal binary partitioning tree splitting mode in accordance with some implementations of the present disclosure.

[0016] FIG. 3D is schematic diagram illustrating vertical ternary partitioning tree splitting mode in accordance with some implementations of the present disclosure.

[0017] FIG. 3E is schematic diagram illustrating horizontal ternary partitioning tree splitting mode in accordance with some implementations of the present disclosure.

[0018] FIG. 4 illustrates a simple FC-NN consisting of input layer, output layer, and multiple hidden layers in accordance with some implementations of the present disclosure.

[0019] FIG. 5A illustrates an FC-NN with two hidden layers in accordance with some implementations of the present disclosure.

[0020] FIG. 5B illustrates an example of CNN in which the dimension of the second hidden layer is [W, H, Depth] in accordance with some implementations of the present disclosure.

[0021] FIG. 6 illustrates an example of applying spatial filters with an input image in accordance with some implementations of the present disclosure.

[0022] FIG. 7A illustrates a single image super-resolution (ResNet) including a residual block as the element of ResNet that is elementwise added with its input by identity connection in accordance with some implementations of the present disclosure.

[0023] FIG. 7B illustrates an example of ResNet by staking residual modules in accordance with some implementations of the present disclosure.

[0024] FIG. 8 illustrates a hybrid training framework in accordance with some implementations of the present disclosure.

[0025] FIG. 9A illustrates an example of online training on an offline trained network and a simple network connected to the offline trained network in accordance with some implementations of the present disclosure.

[0026] FIG. 9B illustrates an example of online training on an offline trained network in accordance with some implementations of the present disclosure.

[0027] FIG. 9C illustrates an example of online training on an offline trained network and a simple network connected to the offline trained network in accordance with some implementations of the present disclosure.

[0028] FIG. 9D illustrates an example of a residual block included in an offline trained network in accordance with some implementations of the present disclosure.

[0029] FIG. 10 is a block diagram illustrating an apparatus for hybrid training on neural networks for video coding in accordance with some implementations of the present disclosure.

[0030] FIG. 11 is a flowchart illustrating a process for hybrid training on neural networks for video coding in accordance with some implementations of the present disclosure.

[0031] FIG. 12 is a block diagram illustrating a system for encoding and decoding video blocks in accordance with some examples of the present disclosure.

DETAILED DESCRIPTION

[0032] Reference will now be made in detail to specific implementations, examples of which are illustrated in the accompanying drawings. In the following detailed description, numerous non-limiting specific details are set forth in order to assist in understanding the subject matter presented herein. But it will be apparent to one of ordinary skill in the art that various alternatives may be used. For example, it will be apparent to one of ordinary skill in the art that the subject matter presented herein can be implemented on many types of electronic devices with digital video capabilities.

[0033] Reference throughout this specification to “one embodiment,” “an embodiment,” “an example,” “some embodiments,” “some examples,” or similar language means that a particular

feature, structure, or characteristic described is included in at least one embodiment or example. Features, structures, elements, or characteristics described in connection with one or some embodiments are also applicable to other embodiments, unless expressly specified otherwise.

[0034] Throughout the disclosure, the terms "first," "second," "third," etc. are all used as nomenclature only for references to relevant elements, e.g., devices, components, compositions, steps, etc., without implying any spatial or chronological orders, unless expressly specified otherwise. For example, a "first device" and a "second device" may refer to two separately formed devices, or two parts, components, or operational states of a same device, and may be named arbitrarily.

[0035] The terms "module," "sub-module," "circuit," "sub-circuit," "circuitry," "sub-circuitry," "unit," or "sub-unit" may include memory (shared, dedicated, or group) that stores code or instructions that can be executed by one or more processors. A module may include one or more circuits with or without stored code or instructions. The module or circuit may include one or more components that are directly or indirectly connected. These components may or may not be physically attached to, or located adjacent to, one another.

[0036] As used herein, the term "if" or "when" may be understood to mean "upon" or "in response to" depending on the context. These terms, if appear in a claim, may not indicate that the relevant limitations or features are conditional or optional. For example, a method may comprise steps of: i) when or if condition X is present, function or action X' is performed, and ii) when or if condition Y is present, function or action Y' is performed. The method may be implemented with both the capability of performing function or action X', and the capability of performing function or action Y'. Thus, the functions X' and Y' may both be performed, at different times, on multiple executions of the method.

[0037] A unit or module may be implemented purely by software, purely by hardware, or by a combination of hardware and software. In a pure software implementation, for example, the unit or module may include functionally related code blocks or software components, that are directly or indirectly linked together, so as to perform a particular function.

[0038] FIG. 12 is a block diagram illustrating an exemplary system 10 for encoding and decoding video blocks in parallel in accordance with some implementations of the present disclosure. As shown in FIG. 12, the system 10 includes a source device 12 that generates and encodes video data to be decoded at a later time by a destination device 14. The source device 12

and the destination device 14 may comprise any of a wide variety of electronic devices, including desktop or laptop computers, tablet computers, smart phones, set-top boxes, digital televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, or the like. In some implementations, the source device 12 and the destination device 14 are equipped with wireless communication capabilities.

[0039] In some implementations, the destination device 14 may receive the encoded video data to be decoded via a link 16. The link 16 may comprise any type of communication medium or device capable of moving the encoded video data from the source device 12 to the destination device 14. In one example, the link 16 may comprise a communication medium to enable the source device 12 to transmit the encoded video data directly to the destination device 14 in real time. The encoded video data may be modulated according to a communication standard, such as a wireless communication protocol, and transmitted to the destination device 14. The communication medium may comprise any wireless or wired communication medium, such as a Radio Frequency (RF) spectrum or one or more physical transmission lines. The communication medium may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication medium may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from the source device 12 to the destination device 14.

[0040] In some other implementations, the encoded video data may be transmitted from an output interface 22 to a storage device 32. Subsequently, the encoded video data in the storage device 32 may be accessed by the destination device 14 via an input interface 28. The storage device 32 may include any of a variety of distributed or locally accessed data storage media such as a hard drive, Blu-ray discs, Digital Versatile Disks (DVDs), Compact Disc Read-Only Memories (CD-ROMs), flash memory, volatile or non-volatile memory, or any other suitable digital storage media for storing the encoded video data. In a further example, the storage device 32 may correspond to a file server or another intermediate storage device that may hold the encoded video data generated by the source device 12. The destination device 14 may access the stored video data from the storage device 32 via streaming or downloading. The file server may be any type of computer capable of storing the encoded video data and transmitting the encoded video data to the destination device 14. Exemplary file servers include a web server (e.g., for a website), a File Transfer Protocol (FTP) server, Network Attached Storage (NAS) devices, or a

local disk drive. The destination device 14 may access the encoded video data through any standard data connection, including a wireless channel (e.g., a Wireless Fidelity (Wi-Fi) connection), a wired connection (e.g., Digital Subscriber Line (DSL), cable modem, etc.), or a combination of both that is suitable for accessing encoded video data stored on a file server. The transmission of the encoded video data from the storage device 32 may be a streaming transmission, a download transmission, or a combination of both.

[0041] As shown in FIG. 12, the source device 12 includes a video source 18, a video encoder 20 and the output interface 22. The video source 18 may include a source such as a video capturing device, e.g., a video camera, a video archive containing previously captured video, a video feeding interface to receive video from a video content provider, and/or a computer graphics system for generating computer graphics data as the source video, or a combination of such sources. As one example, if the video source 18 is a video camera of a security surveillance system, the source device 12 and the destination device 14 may form camera phones or video phones. However, the implementations described in the present application may be applicable to video coding in general, and may be applied to wireless and/or wired applications.

[0042] The captured, pre-captured, or computer-generated video may be encoded by the video encoder 20. The encoded video data may be transmitted directly to the destination device 14 via the output interface 22 of the source device 12. The encoded video data may also (or alternatively) be stored onto the storage device 32 for later access by the destination device 14 or other devices, for decoding and/or playback. The output interface 22 may further include a modem and/or a transmitter.

[0043] The destination device 14 includes the input interface 28, a video decoder 30, and a display device 34. The input interface 28 may include a receiver and/or a modem and receive the encoded video data over the link 16. The encoded video data communicated over the link 16, or provided on the storage device 32, may include a variety of syntax elements generated by the video encoder 20 for use by the video decoder 30 in decoding the video data. Such syntax elements may be included within the encoded video data transmitted on a communication medium, stored on a storage medium, or stored on a file server.

[0044] In some implementations, the destination device 14 may include the display device 34, which can be an integrated display device and an external display device that is configured to communicate with the destination device 14. The display device 34 displays the decoded video

data to a user, and may comprise any of a variety of display devices such as a Liquid Crystal Display (LCD), a plasma display, an Organic Light Emitting Diode (OLED) display, or another type of display device.

[0045] The video encoder 20 and the video decoder 30 may operate according to proprietary or industry standards, such as VVC, HEVC, MPEG-4, Part 10, AVC, or extensions of such standards. It should be understood that the present application is not limited to a specific video encoding/decoding standard and may be applicable to other video encoding/decoding standards. It is generally contemplated that the video encoder 20 of the source device 12 may be configured to encode video data according to any of these current or future standards. Similarly, it is also generally contemplated that the video decoder 30 of the destination device 14 may be configured to decode video data according to any of these current or future standards.

[0046] The video encoder 20 and the video decoder 30 each may be implemented as any of a variety of suitable encoder and/or decoder circuitry, such as one or more microprocessors, Digital Signal Processors (DSPs), Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. When implemented partially in software, an electronic device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the video encoding/decoding operations disclosed in the present disclosure. Each of the video encoder 20 and the video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective device.

[0047] Like HEVC, VVC is built upon the block-based hybrid video coding framework. FIG. 1 is a block diagram illustrating a block-based video encoder in accordance with some implementations of the present disclosure. In the encoder 100, the input video signal is processed block by block, called coding units (CUs). The encoder 100 may be the video encoder 20 as shown in FIG. 12. In VTM-1.0, a CU can be up to 128x128 pixels. However, different from the HEVC which partitions blocks only based on quad-trees, in VVC, one coding tree unit (CTU) is split into CUs to adapt to varying local characteristics based on quad/binary/ternary-tree. Additionally, the concept of multiple partition unit type in the HEVC is removed, i.e., the separation of CU, prediction unit (PU) and transform unit (TU) does not exist in the VVC anymore; instead, each CU is always used as the basic unit for both prediction and transform

without further partitions. In the multi-type tree structure, one CTU is firstly partitioned by a quad-tree structure. Then, each quad-tree leaf node can be further partitioned by a binary and ternary tree structure.

[0048] FIGS. 3A-3E are schematic diagrams illustrating multi-type tree splitting modes in accordance with some implementations of the present disclosure. FIGS. 3A-3E respectively show five splitting types including quaternary partitioning (FIG. 3A), vertical binary partitioning (FIG. 3B), horizontal binary partitioning (FIG. 3C), vertical ternary partitioning (FIG. 3D), and horizontal ternary partitioning (FIG. 3E).

[0049] For each given video block, spatial prediction and/or temporal prediction may be performed. Spatial prediction (or "intra prediction") uses pixels from the samples of already coded neighboring blocks (which are called reference samples) in the same video picture/slice to predict the current video block. Spatial prediction reduces spatial redundancy inherent in the video signal. Temporal prediction (also referred to as "inter prediction" or "motion compensated prediction") uses reconstructed pixels from the already coded video pictures to predict the current video block. Temporal prediction reduces temporal redundancy inherent in the video signal. Temporal prediction signal for a given CU is usually signaled by one or more motion vectors (MVs) which indicate the amount and the direction of motion between the current CU and its temporal reference. Also, if multiple reference pictures are supported, one reference picture index is additionally sent, which is used to identify from which reference picture in the reference picture store the temporal prediction signal comes.

[0050] After spatial and/or temporal prediction, an intra/inter mode decision circuitry 121 in the encoder 100 chooses the best prediction mode, for example based on the rate-distortion optimization method. The block predictor 120 is then subtracted from the current video block; and the resulting prediction residual is de-correlated using the transform circuitry 102 and the quantization circuitry 104. The resulting quantized residual coefficients are inverse quantized by the inverse quantization circuitry 116 and inverse transformed by the inverse transform circuitry 118 to form the reconstructed residual, which is then added back to the prediction block to form the reconstructed signal of the CU. Further, in-loop filtering 115, such as a deblocking filter, a sample adaptive offset (SAO), and/or an adaptive in-loop filter (ALF) may be applied on the reconstructed CU before it is put in the reference picture store of the picture buffer 117 and used to code future video blocks. To form the output video bitstream 114, coding mode (inter or

intra), prediction mode information, motion information, and quantized residual coefficients are all sent to the entropy coding unit 106 to be further compressed and packed to form the bit-stream.

[0051] For example, a deblocking filter is available in AVC, HEVC as well as the now-current version of VVC. In HEVC, an additional in-loop filter called SAO is defined to further improve coding efficiency. In the now-current version of the VVC standard, yet another in-loop filter called ALF is being actively investigated, and it has a good chance of being included in the final standard.

[0052] These in-loop filter operations are optional. Performing these operations helps to improve coding efficiency and visual quality. They may also be turned off as a decision rendered by the encoder 100 to save computational complexity.

[0053] It should be noted that intra prediction is usually based on unfiltered reconstructed pixels, while inter prediction is based on filtered reconstructed pixels if these filter options are turned on by the encoder 100.

[0054] FIG. 2 is a block diagram illustrating a block-based video decoder 200 which may be used in conjunction with many video coding standards. This decoder 200 is similar to the reconstruction-related section residing in the encoder 100 of FIG. 1. The block-based video decoder 200 may be the video decoder 30 as shown in FIG. 12. In the decoder 200, an incoming video bitstream 201 is first decoded through an Entropy Decoding 202 to derive quantized coefficient levels and prediction-related information. The quantized coefficient levels are then processed through an Inverse Quantization 204 and an Inverse Transform 206 to obtain a reconstructed prediction residual. A block predictor mechanism, implemented in an Intra/inter Mode Selector 212, is configured to perform either an Intra Prediction 208, or a Motion Compensation 210, based on decoded prediction information. A set of unfiltered reconstructed pixels are obtained by summing up the reconstructed prediction residual from the Inverse Transform 206 and a predictive output generated by the block predictor mechanism, using a summer 214.

[0055] The reconstructed block may further go through an In-Loop Filter 209 before it is stored in a Picture Buffer 213 which functions as a reference picture store. The reconstructed video in the Picture Buffer 213 may be sent to drive a display device, as well as used to predict future

video blocks. In situations where the In-Loop Filter 209 is turned on, a filtering operation is performed on these reconstructed pixels to derive a final reconstructed Video Output 222.

[0056] When neural networks are utilized in a video coding system, the neural network models may be offline or online trained. In the present disclosure, methods and apparatus of a hybrid training framework are described in which both online and offline training methods are used to improve the video coding efficiency. For easier illustration, the present disclosure takes in-loop filtering as an example, while the proposed ideas may be extended to other functioning parts of a video coding system as well. For example, prediction mode decision or transform in FIG. 1. The proposed hybrid training methods may be applied to the traditional block-based video coding system or end-to-end neural network-based video coding systems. In one or more embodiments, the proposed neural network and the related hybrid training methods may be applied to the traditional block-based video coding system by improving or replacing one or more traditional tools. For example, the traditional tools may include the existing traditional in-loop filters such as luma mapping with chroma scaling, deblocking filter, sample adaptive offset, adaptive loop filter and cross-component adaptive loop filter. These traditional tools may be partially or completely configured or conditionally turned on/off when the proposed neural network is applied.

[0057] In other embodiments, the proposed neural network and the related hybrid training methods may be applied to end-to-end neural network-based video coding systems by adding new neural network modules/blocks or improving or replacing one or more existing neural network modules/blocks. For example, the existing neural network modules/blocks in an end-to-end neural network-based video coding system may include deep intra coding, deep inter coding, deep residual coding or/and deep entropy coding. The proposed neural network may be applied as a new supplementary network and placed anywhere between the existing modules/blocks. Alternatively or additionally, the proposed neural network may be applied to replace the existing modules/blocks by partially or completely configuring or conditionally turning on/off these existing modules/blocks.

[0058] The neural network techniques, e.g., fully connected neural network (FC-NN), convolutional neural network (CNN), and long short-term memory network (LSTM), have already achieved significant success in many research domains, including computer vision and video understanding.

Fully-connected Neural Network (FC-NN)

[0059] FIG. 4 illustrates a simple FC-NN consisting of input layer, output layer, and multiple hidden layers in accordance with some implementations of the present disclosure. At k-th layer, the output $f^k(x^{k-1}, W^k, B^k)$, is generated by

$$f^k(x^{k-1}, W^k, B^k) = \delta(x^{k-1} * W^k + B^k) \quad (1)$$

$$x^{k-1} * W^k + B^k = \left[x_1^{k-1}, \ldots, x_j^{k-1}, \ldots, x_M^{k-1}\right] \cdot \begin{bmatrix} W_{1,1}^k & \cdots & W_{0,N}^k \\ \vdots & \ddots & \vdots \\ W_{M,1}^k & \cdots & W_{M,N}^k \end{bmatrix} + \left[B_1^{k-1}, \ldots, B_j^{k-1}, \ldots, B_M^{k-1}\right] \quad (2)$$

where $x^{k-1} \in R^M$ is the output of (k-1)-th layer, $W^k \in R^{M*N}$ and $B^k \in R^N$ are the weight and the bias at k-th layer. $\delta(\cdot)$ is the activation function, e.g., the Rectified Linear Unit (ReLU) function as defined in Eq. (3).

$$\delta(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3)$$

Therefore, the general form of a K-layer FC-NN is written as

$$FCNN(x) = f^K(\ldots f^k(f^{k-1}(\ldots f^1(x, W^1, B^1)\ldots), W^k, B^k)\ldots, W^K, B^K), \ for\ 1 \leq k \leq K \quad (4)$$

[0060] According to the universal approximation hypothesizes and Eq. (4), given any continuous function $g(x)$ and some $\varepsilon > 0$, there exists a neural network $f(x)$ with a reasonable choice of non-linearity e.g., ReLU, such that $\forall x, |g(x) - f(x)| < \varepsilon$. Therefore, many empirical studies applied neural network as an approximator to mimic a model with hidden variables in order to extract explainable features under the surfaces. For example, applying in image recognition, FC-NN helps researchers to construct a system that understands not just a single pixel, but increasingly much deeper and complex sub-structures, e.g., edges, textures, geometric shapes, and objects.

Convolutional Neural Network (CNN)

[0061] FIG. 5A illustrates an FC-NN with two hidden layers in accordance with some implementations of the present disclosure. CNN, a popular neural network architecture for image or video applications, is very similar to the FC-NN as shown in FIG. 5A, which includes weights and bias metrices. A CNN can be seen as a 3-D version of neural network. FIG. 5B illustrates an example of CNN in which the dimension of the second hidden layer is [W, H, Depth] in accordance with some implementations of the present disclosure. In FIG. 5B, neurons are arranged in 3-Dimensional structure (width, height, and depth) to form a CNN, and the second

hidden layer is visualized. In this example, the input layer holds input image or video frames therefore its width and height are same as input data. To apply with image or video applications, each neuron in CNN is a spatial filter element with extended depth aligned with its input, e.g., the depth is 3 if there are 3 color components in input images.

[0062] FIG. 6 illustrates an example of applying spatial filters with an input image in accordance with some implementations of the present disclosure. As shown in FIG. 6, the dimension of basic element in CNN is defined as $[Filter_{width}, Filter_{height}, Input_{depth}, Output_{depth}]$ and set to [5, 5, 3, 4] in this example. Each spatial filter performs 2-dimensional spatial convolution with 5*5*3 weights on an input image. The input image may be a 64x64x3 image. Then, 4 convolutional results are outputted. Therefore, the dimension of filtered results is [64+4, 64+4, 4] if padding the boundary with additional 2 pixels.

Residual Network (ResNet)

[0063] In image classification, the accuracy is saturated and degrades rapidly when the depth of neural network increases. To be more specifically, adding more layers on deep neural network results in higher training error because the gradient is gradually vanishing along the deep network and toward to zero gradient at the end. Then, the ResNet composed of residual blocks comes to resolve the degradation problem by introducing the identity connection.

[0064] FIG. 7A illustrates a ResNet including a residual block as the element of ResNet that is elementwise added with its input by identity connection in accordance with some implementations of the present disclosure. As shown in FIG. 7A, a basic module of ResNet is consist of the residual block and the identity connection. According to the universal approximation hypothesizes, given an input x, weighted layers with activation function in residual block approximate a hidden function $F(x)$ rather than the output $H(x) = F(x) + x$.

[0065] By stacking non-linear multi-layer neural network, the residual block explores the features that represent the local characteristic of input images. Without introducing neither additional parameters and computational complexity, the identity connection is proven to make deep learning network trainable by skip one or more non-linear weighted layers as shown in FIG. 7A. Skipping weighted layers, the differential output of the residual layers can be written as

$$\frac{\partial H(x)}{\partial x} = \frac{\partial F(x)}{\partial x} + 1 \qquad (5)$$

[0066] Therefore, even if the differential term $\frac{\partial F(x)}{\partial x}$ is gradually decreasing toward zero, the identity term can still carry on and pass the input to next layer instead of stuck at zero gradient as well as blocking information propagation. If a neuron cannot propagate information to next neuron, it is seen as dead neuron, which is non-trainable element in neural network. After addition, another non-linear activation function can be applied as well. FIG. 7B illustrates an example of ResNet by staking residual modules in accordance with some implementations of the present disclosure. As shown in FIG. 7B, the residual features are fused with the identity features before propagating to the next module.

[0067] For better generalization, offline trained neural networks may include multiple layers and are iteratively updated during the training stage. The relatively large number of layers may lead to a big parameter size such that excessive processing delay and memory consumption could be observed at the inference stage. In addition, the offline trained neural network models are learned from a given data set, which may not reflect actual data characteristics, e.g., illumination, texture, object deformations, etc., when the models are applied elsewhere.

[0068] On the other hand, online trained neural networks adapt well to real application scenarios, but may be constrained with a smaller parameter size, leading to a limited utilization of the non-linear representation ability of a neural network.

[0069] For video coding systems, the ground truth, e.g., uncompressed video data, is always accessible at the encoder side, which makes online training viable. In order to explore the combined benefits, a hybrid training framework including both online and offline training are disclosed.

Hybrid Training Framework

[0070] In the present disclosure, a hybrid training framework is proposed to improve the coding efficiency of the neural networks utilized for a video coding system. The proposed hybrid training framework may include any combination of below features.

[0071] In some examples, one or multiple neural networks with the same or different network architecture may be offline trained first. The data sets selected for the offline training may include different data characteristics, e.g., still images with various background textures, videos with low-motion objects, etc., for better generalization purpose.

[0072] Then the offline trained models are refined by online training, e.g., online trained for adapting to specific video content. In this online training stage, a small number of layers may be

updated only. For the layers not being updated, for example, the gradients of those layers may be forcefully set to zero if the training algorithm is gradient descent based. On a specific neural network framework, the layers not being updated may be set to not trainable, e.g., PyTorch.

[0073] In some other examples, in the online training stage, the offline trained models may be treated as feature extractors with or without further refinement, and a new simple network with very few layers or weights may be connected after the offline trained models.

[0074] At different training stages, the same or different learning algorithms may be used. In one or more examples, at the offline training stage, where the speed of convergence is not stringent and generalization is more important, learning algorithms such as stochastic gradient descent (SGD) may be preferably used. On the other hand, at the online training stage, where both the training speed and fast fitting to the training data are critical, fast learning algorithms such as adaptive moment estimation (ADAM) may be favorably used.

[0075] In one or more examples, the switch between different algorithms may be performed within the same training stage. For example, at the offline training stage, in the beginning epochs, fast algorithms such as ADAM is used, then the learning algorithm is switched to be SGD when the loss function generates a loss below a predefined threshold value.

[0076] In some examples, when learning algorithms are switched either within the same training stage or between different training stage, a suitable starting value of the learning rate parameter may be switched as well. The specific value of the learning rate may be associated with the loss function or the algorithm itself.

[0077] For online training-based model refinement, two factors related to temporal scope include temporal frequency and temporal coverage. Temporal frequency determines when the online training-based model refinement is performed.

[0078] In one or more examples, the online training may be performed periodically, such as every two seconds, every one second, or every N video frames, where N may be a positive integer. In some examples, the online training may be triggered based on event.

[0079] In one or more examples, the online training may be triggered by the video encoder behaviors, for example, generation of an IDR (Instantaneous Decoder Refresh) coded picture. Every time a new IDR frame is generated, the subsequent motion predicted video frames may contain totally different level of artifacts, e.g., due to the change of the reference pictures.

[0080] In one or more examples, the online training can be triggered by the video encoder configurations, for example, intra period value or frame rate. The encoder configuration may be dynamically changed due to different reasons, such as bit rate control, error concealment, etc. The change of the encoder configuration may be activated by a predefined profile or the lower protocol layers of the operation systems running on the device.

[0081] In one or more examples, the online training may be triggered by the video content, for example, scene detection. When the objects, the background of the objects, or the objects and the background of the objects within a video frame is changed, the prediction, partition granularity within a video frame, or the prediction and partition granularity within the video frame is likely to be changed as well. These content changes are the main reasons where the video coding bits, coding distortions, or the video coding bits and coding distortions come from.

[0082] In some examples, the online training may be triggered in a manner of a combination of periodic and event triggering. For example, the online training may be performed periodically first and then skipped in response to a triggering event.

[0083] Furthermore, temporal coverage determines what training data are used for the model update. Speed of convergence is critical for online training. Training data not only impacts the processing delay but also the training convergence, e.g., highly diverse video data is difficult for the training to converge.

[0084] In one or more examples, the coding information such as temporal layer information may be used as reference to select training data. For example, within each group of pictures, one picture from each temporal layer may be selected for online training.

[0085] In one or more examples, the video content may be used as reference to select training data. For example, within each online training period, adjacent video frames may likely have high temporal redundancy, which means the objects and background are highly similar. In this case, scene information or regions of interest including both objects and background textures can be extracted at the encoder side, and these extracted small patches instead of the whole pictures can be used as the online training data.

[0086] In some examples, information other than video frames are also used as training data, e.g., for multi-frame-based training. For example, the motion information between each selected video frame and the anchor frame, e.g., reference pictures, may be used for training the motion

predicted frames. In this case, optical flow map may be generated as the representation of motion information.

[0087] At the online training stage, each selected video frame may be split or cropped into small patches before used as online training data. The small patches may be clustered according to certain criteria. In some examples, only the patches classified as the same or different clusters are selected to be online training data.

[0088] In some examples, each small patch may be represented by a scale-invariant feature descriptor, e.g., histogram of oriented gradients or a template, e.g., bag of words, based feature vectors. Different patches are clustered by the representation similarities.

[0089] In some examples, patches are clustered by different level of distortions.

[0090] At the online training stage, the video frames or patches with lower qualities, e.g., quantization parameter is greater than a certain threshold value, may have blurred object boundaries or texture details. Therefore, those data may cause severe training confusions which make the training hard to converge and should not be selected as training data.

[0091] For video coding, offline trained neural network models are supposed to work on a variety of diverse video sequences after training thus require good generalization, while online trained neural network models need only fit the video data within a small temporal space, e.g., fits the video frames within an intra period, thus require good overfitting. Due to different requirements, the model training process can be considered in the context of the generalization-overfitting trade-off.

[0092] In one or more examples, the balance of the generalization-overfitting trade-off can be controlled by training the network on a sufficient amount of data. This has already been discussed above. For example, offline models are trained on larger and diverse data sets, while online models or model refinement are trained within a constrained temporal scope, spatial scope and quality scope.

[0093] In one or more examples, the balance of the generalization-overfitting trade-off can be controlled by changing the network complexity. In one example, the network complexity can be changed by adjusting the network structure. Specifically, the number of trainable model parameters or weights can be flexibly configured. For example, online trained networks may set only a small number of layers to be trainable, while offline trained networks may set a large

number of layers or all layers to be trainable. This method can be termed as structural stabilization.

[0094] In another example, the network complexity can be changed by adjusting the dynamic range of network parameters. Parameters with small dynamic range are less sensitive to statistical fluctuations in the input training data, thus has better generalization capability. However, if overfitting is desirable, such as the online models, parameters with larger dynamical range are preferable.

[0095] In some examples, data type, e.g., integer or floating point, is used to control the dynamical range of network parameters or weight. In some other examples, regularization methods are used to control the dynamical range of network parameters or weight. One typical regularization method is to penalize the model, e.g., loss function, during training process based on the magnitude of the parameters or weights value. This approach is called weight regularization or weight decay.

[0096] FIG. 8 illustrates a hybrid training framework in accordance with some implementations of the present disclosure. As shown in the FIG. 8, one or multiple models are first offline trained with large amount of image or video data which have diverse data characteristics, then the models are repeatedly refined by online training with small amount of data. The purpose of the online training is to reduce the network complexity such that the network may better overfit the small amount of data within each refinement period. The network complexity can be reduced by limiting the number of trainable layers, parameters, or weights within the network, or by setting a suitable dynamic range of those trainable parameters or weights.

[0097] FIGS. 9A-9D illustrate examples on how to perform online training on top of an offline trained network and/or a simple network connected to the offline trained network. As shown in the FIGS. 9A-9D, a new simple network connected to the offline trained network may be trained at the online training stage as shown in FIG. 9A, or a subset of layers within the offline trained network are refined as shown in FIG. 9B, or both the new simple network and a subset of layers within the offline trained network are trained at the online training stage as shown in FIG. 9C.

Transmission of the Online Trained Parameters or Weights

[0098] The output after each online training is a limited size of network parameters or weights. To reduce the signaling overhead, the parameters or weights need to be compressed before

written into the bitstream. The compression method may be loss-less, lossy, or a combination of loss-less compression and lossy compression.

[0099] In one or more examples, even the updated parameters may be large and sensitive to the fluctuations of the input data due to overfitting, parameter pruning and quantization can still be performed. However, in order to maintain the sensitivity, manual setup of the sensitivity is beneficial and doable because the number of layers are typically small.

[0100] In one or more examples, instead of directly signaling the updated parameters, mean-shift or weight sharing like methods can be used to as preprocessing step. Also, the difference before and after the parameter update may be derived and constructed as a differential map, which can further reduce the bits to signal the updated parameters.

Interactions with Other Existing In-Loop Filters

[0101] If the networks trained for the video codec are used as an in-loop filter, the updated parameters after online training may be used to guide the training of other in-loop filters. For example, for adaptive in-loop filter (ALF), the derived filter coefficients may be extended or tailored to approximate the updated parameters. Once ALF filter coefficients are refined, the updated neural network parameters may not need to be signaled, not only saving bits but also reduce the processing delay due to using the neural network model filtering.

Constrained Online Training Process

[0102] Considering the wide variety of input video contents and the limitation of the training data set used for offline training, online training based update of the parameters/weights is proposed in previous sections. As shown in the FIGS. 9A-9D, the update may be performed on a newly added small network as shown in FIG. 9A, or on a few layers within the offline trained network as shown in FIG. 9B, or both as shown in FIG. 9C. This type of online training is to restrict the training complexity as well as the number of updated parameters/weights from the network structure perspective.

[0103] In one aspect of the present disclosure, regularization based constraint is proposed for online training process. In some examples, regularization is applied using a dynamic range. In one or more examples, if the trainable parameters/weights in the online training process aim to update or refine an offline trained network, the dynamic range control is applied to the parameter/weight difference before and after the online training. For example, for any parameter/weight, the absolute value is X before online training. During the online training

process, the dynamic range may be regularized as Y. In this case, any parameter/weight that is updated during online training is clipped by the range of [X-Y, X+Y]. After the clipping, this parameter may be updated/refined again by a new round of online training process or fixed at this value without further update.

[0104] In one or more examples, if the trainable parameters/weights are from a newly added small network, which is not trained during the offline process, the dynamic range control is applied to the parameter/weight itself. For example, for any parameter/weight, the value after online training is clipped by a pre-defined range such as [-R, R], where R is the maximum allowed limit. After the clipping, this parameter may be updated/refined again by a new round of online training process or fixed at this value without further update.

[0105] In some examples, regularization is applied using a parameter type. When online training is performed, the parameters/weights within a trainable neural layer may be updated differently. In one or more examples, the parameters/weights within a trainable neural layer are classified as different categories.

[0106] In one or more examples, the classification may be dependent on the intensity value of each parameter/weight. In this case, different parameters/weights may be assigned to different bands, where each band is a range of intensity value. For example, two bands may be predefined, and the absolute value below a certain threshold is one band, and above the threshold is another band.

[0107] In one or more examples, the classification may be dependent on the role in a neuron, e.g., weights and biases.

[0108] Furthermore, based on a specified classification method, the parameters/weights in different classes may be updated differently. In one or more examples, the parameters may be divided into two or more bands based on the absolute values of the parameters. The two or more bands may include a high band and a low band. For example, the parameters in the high band (the range in this band is in a high end) may be updated with larger dynamic range, while the parameters in the low band (the range in this band is in a low end) may be updated within smaller dynamic range. For example, when a dynamic range is 0, it means the update is disabled, or this parameter is not updated/refined.

[0109] In some examples, the parameters in some categories may be fixed during the online training. For example, to reduce the weight update signaling, only the bias terms of a neural layer

are updated, while the weights are not. Alternatively, only the weights or biases with larger value may be updated during the online training.

[0110] In some examples, regularization is applied using a parameter location. For example, when a neural network layer is updated during online training, this layer may have multiple channels. In one or more examples, a subset of channels may be selected as trainable during the online training process.

[0111] In some examples, regularization is applied using a loss function. During the online training, the trainable parameters/weights are updated or trained based on the loss function. In some examples, if the trainable parameters/weights are from the offline trained neural layers, the difference of the parameters/weights between and after each round of online training epoch is used to generate a penalty term, which can be added on the existing loss function to encourage smaller difference after update. An epoch may indicate training the neural network with all the training data for one cycle.

[0112] In some other examples, if the trainable parameters/weights are from the newly added neural layers, the value of the parameters/weights is used to generate a penalty term, which can be added on the existing loss function to encourage smaller parameters/weights.

[0113] In another aspect of the present disclosure, post-processing based constraint is proposed for online training process. In some examples, the post-processing based constraint may be implemented using value approximation. Once online training process is finished, a subset of the trainable parameters/weights may be updated or newly trained with a new value.

[0114] In some examples, for the updated/refined parameters/weights, which are previously trained in offline stage, the update difference of each parameter/weight may be quantized before the compression. Alternatively, the update difference of one parameter/weight may be approximated to a specific value, such as to the power of 2. In another example, when the update difference is below a threshold, the update difference is set to 0.

[0115] In some examples, regularization of the update difference may be implemented using a look up table (LUT). Instead of directly signaling the values of update difference, a set of LUT indexed values may be signaled. In one example, a set of quantized or not quantized update difference values may be represented as (x1, x2, …. xn), where the difference between x1 and xn represents the largest update difference value may be allowed, and the index values are represented as (1, …., n).

[0116] In some examples, the updated/refined parameters/weights which are previously trained in offline stage may be classified based on parameter types. Accordingly, the update differences corresponding to parameters in difference classes or bands may be approximated differently.

[0117] In other examples, for the newly trained parameters/weights, which are from a new network only trained during online process, the value of each parameter/weight may be quantized before the compression. Alternatively, the value of one parameter/weight may be approximated to a specific value, such as to the power of 2. In another example, when the value of one parameter/weight is below a threshold, the value is set to 0.

[0118] In some examples, regularization of the newly trained parameters/weights may be implemented using an LUT. The implementation of the LUT in this case may be similarly designed as the update difference for updated/refined parameters/weights.

[0119] In some examples, the newly trained parameters/weights from the new network only trained during online process may be classified based on parameter types. Accordingly, the values of parameters/weights in different classes or bands may be approximated differently.

[0120] It is proposed that all above proposed techniques may be used in any combination.

[0121] FIG. 10 is a block diagram illustrating an apparatus for hybrid training on neural networks for video coding in accordance with some implementations of the present disclosure. The apparatus 1000 may be a terminal, such as a mobile phone, a tablet computer, a digital broadcast terminal, a tablet device, or a personal digital assistant.

[0122] As shown in FIG. 10, the apparatus 1000 may include one or more of the following components: a processing component 1002, a memory 1004, a power supply component 1006, a multimedia component 1008, an audio component 1010, an input/output (I/O) interface 1012, a sensor component 1014, and a communication component 1016.

[0123] The processing component 1002 usually controls overall operations of the apparatus 1000, such as operations relating to display, a telephone call, data communication, a camera operation, and a recording operation. The processing component 1002 may include one or more processors 1020 for executing instructions to complete all or a part of steps of the above method. Further, the processing component 1002 may include one or more modules to facilitate interaction between the processing component 1002 and other components. For example, the processing component 1002 may include a multimedia module to facilitate the interaction between the multimedia component 1008 and the processing component 1002.

[0124] The memory 1004 is configured to store different types of data to support operations of the apparatus 1000. Examples of such data include instructions, contact data, phonebook data, messages, pictures, videos, and so on for any application or method that operates on the apparatus 1000. The memory 1004 may be implemented by any type of volatile or non-volatile storage devices or a combination thereof, and the memory 1004 may be a Static Random Access Memory (SRAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), an Erasable Programmable Read-Only Memory (EPROM), a Programmable Read-Only Memory (PROM), a Read-Only Memory (ROM), a magnetic memory, a flash memory, a magnetic disk or a compact disk.

[0125] The power supply component 1006 supplies power for different components of the apparatus 1000. The power supply component 1006 may include a power supply management system, one or more power supplies, and other components associated with generating, managing and distributing power for the apparatus 1000.

[0126] The multimedia component 1008 includes a screen providing an output interface between the apparatus 1000 and a user. In some examples, the screen may include a Liquid Crystal Display (LCD) and a Touch Panel (TP). If the screen includes a touch panel, the screen may be implemented as a touch screen receiving an input signal from a user. The touch panel may include one or more touch sensors for sensing a touch, a slide and a gesture on the touch panel. The touch sensor may not only sense a boundary of a touching or sliding actions, but also detect duration and pressure related to the touching or sliding operation. In some examples, the multimedia component 1008 may include a front camera and/or a rear camera. When the apparatus 1000 is in an operation mode, such as a shooting mode or a video mode, the front camera and/or the rear camera may receive external multimedia data.

[0127] The audio component 1010 is configured to output and/or input an audio signal. For example, the audio component 1010 includes a microphone (MIC). When the apparatus 1000 is in an operating mode, such as a call mode, a recording mode and a voice recognition mode, the microphone is configured to receive an external audio signal. The received audio signal may be further stored in the memory 1004 or sent via the communication component 1016. In some examples, the audio component 1010 further includes a speaker for outputting an audio signal.

[0128] The I/O interface 1012 provides an interface between the processing component 1002 and a peripheral interface module. The above peripheral interface module may be a keyboard, a click

wheel, a button, or the like. These buttons may include but not limited to, a home button, a volume button, a start button, and a lock button.

[0129] The sensor component 1014 includes one or more sensors for providing a state assessment in different aspects for the apparatus 1000. For example, the sensor component 1014 may detect an on/off state of the apparatus 1000 and relative locations of components. For example, the components are a display and a keypad of the apparatus 1000. The sensor component 1014 may also detect a position change of the apparatus 1000 or a component of the apparatus 1000, presence or absence of a contact of a user on the apparatus 1000, an orientation or acceleration/deceleration of the apparatus 1000, and a temperature change of apparatus 1000. The sensor component 1014 may include a proximity sensor configured to detect presence of a nearby object without any physical touch. The sensor component 1014 may further include an optical sensor, such as a CMOS or CCD image sensor used in an imaging application. In some examples, the sensor component 1014 may further include an acceleration sensor, a gyroscope sensor, a magnetic sensor, a pressure sensor, or a temperature sensor.

[0130] The communication component 1016 is configured to facilitate wired or wireless communication between the apparatus 1000 and other devices. The apparatus 1000 may access a wireless network based on a communication standard, such as WiFi, 4G, or a combination thereof. In an example, the communication component 1016 receives a broadcast signal or broadcast related information from an external broadcast management system via a broadcast channel. In an example, the communication component 1016 may further include a Near Field Communication (NFC) module for promoting short-range communication. For example, the NFC module may be implemented based on Radio Frequency Identification (RFID) technology, infrared data association (IrDA) technology, Ultra-Wide Band (UWB) technology, Bluetooth (BT) technology and other technology.

[0131] In an example, the apparatus 1000 may be implemented by one or more of Application Specific Integrated Circuits (ASIC), Digital Signal Processors (DSP), Digital Signal Processing Devices (DSPD), Programmable Logic Devices (PLD), Field Programmable Gate Arrays (FPGA), controllers, microcontrollers, microprocessors, or other electronic elements to perform the above method. A non-transitory computer readable storage medium may be, for example, a Hard Disk Drive (HDD), a Solid-State Drive (SSD), Flash memory, a Hybrid Drive or Solid-

State Hybrid Drive (SSHD), a Read-Only Memory (ROM), a Compact Disc Read-Only Memory (CD-ROM), a magnetic tape, a floppy disk, etc.

[0132] FIG. 11 is a flowchart illustrating a process for hybrid training on neural networks for video coding in accordance with some implementations of the present disclosure.

[0133] In step 1101, the processor 1020 obtains, in an offline training stage, an offline trained network by training a neural network offline. In some examples, the neural network may be trained using a plurality of first data sets. As shown in FIG. 8, the plurality of first data sets may include large offline data, e.g., images, videos.

[0134] In step 1102, the processor 1020 refines, in an online training stage, a plurality of neural network layers with constraint on a plurality of parameters of the plurality of neural network layers. Here, the constraint may represent a set of methods used to minimize the neural network parameter changes before and after the hybrid training. The methods may be various adaptively depending on specific purpose of different experiments in the art. For example, the constraint may include a weight constraint that checks the size of the weights, and if the size exceeds a predefined limit, the weights are rescaled so that their size is below the limit or between a range. In one or more embodiments, depending on where the constraint is applied, there are two types of constraints: regularization-based constraint and post processing based constraint. In the regularization-based constraint, the proposed constraint methods are applied during the online training process. In the post processing based constraint, the proposed constraint methods are applied after the online training process.

[0135] In some examples, the plurality of neural network layers may include at least one neural network layer in the offline trained network or in a simple neural network connected to the offline trained network, as shown in FIGS. 9A-9C. As shown in FIG. 9A, the offline trained network or model includes multiple residual blocks (RBs). Each RB may have the same structure as shown in FIG. 9D. As shown in FIG. 9D, the RB include two convolutional layers and a Rectified Linear Unit (Relu) layer that is sandwiched between the two convolutional layers. A simple network that is connected to the offline trained network may include, but not limited to, two convolutional layers. Online training may be performed on: only the two convolutional layers in the simple network (FIG. 9A), only the last RB in the offline trained network (FIG. 9B), or the last RB in the offline trained network and the two convolutional layers in the simple network (FIG. 9C).

[0136] In some examples, the processor 1020 may repeatedly refine the plurality of neural network layers using a plurality of second data sets. In some examples, the plurality of second data sets may include coding information, video content, and/or information other than video frames, where the coding information may include temporal layer information, the video content may include a plurality of patches extracted based on regions of interest from a video frame, and the information other than video frames may include motion information.

[0137] In some examples, the offline trained network may include more neural network layers than the plurality of neural network layers that are refined, as shown in FIGS. 9A-9C.

[0138] In some examples, the simple neural network may include less neural network layers than the offline trained network.

[0139] In some examples, the plurality of first data sets may include more data than the plurality of second data sets. The plurality of second data sets may be small amount of data as shown in FIG. 8, e.g., a few video frames.

[0140] In some examples, regularization based constraint may be applied on the plurality of parameters of the plurality of neural network layers refined in the online training stage.

[0141] In some examples, the plurality of parameters may be regularized using a dynamic range before and after each online training epoch.

[0142] In some examples, a first parameter of one or more neural network layers in the offline trained network (e.g., trainable RB in the offline trained network) may be clipped by the dynamic range based on a value of the parameters before an online training epoch and a predetermined value. The first parameter may be one parameter of the one or more neural network layers in the offline trained network (e.g., trainable RB in the offline trained network). For example, for any parameter/weight, the absolute value is X before online training. During the online training process, the dynamic range may be regularized as Y. In this case, any parameter/weight that is updated during online training is clipped by the range of [X-Y, X+Y]. After the clipping, this parameter may be updated/refined again by a new round of online training process/epoch or fixed at this value without further update.

[0143] In some examples, a second parameter of the at least one neural network layer in the simple neural network connected to the offline trained network may be clipped by a predefined range. For example, the second parameter may be one trainable parameter/weight from a newly added small network, which is not trained during the offline process, as shown in FIG. 9A or 9B.

The value of the second parameter after online training may be clipped by a pre-defined range such as [-R, R].

[0144] In some examples, a parameter type of the plurality of parameters of neural network layers may be regularized, that is, regularization may be based on parameter types. For examples, the plurality of parameters may be classified into different bands. Parameters in each band may be of a same parameter type. Parameters in different bands may be updated differently.

[0145] In some examples, the plurality of parameters may be classified based on intensity values, roles in neurons (e.g., weights and biases), etc.

[0146] In some examples, different bands may include a first band a second band, where one or more parameters in the first band may be regularized using a first dynamic range and one or more parameters in the second band may be regularized using a second dynamic range, and the first dynamic range may be greater than the second dynamic range.

[0147] For example, the parameters may be divided into two or more bands based on the absolute values of the parameters. The two or more bands may include a high band and a low band. For example, the parameters in the high band (the range in this band is in a high end) may be updated with larger dynamic range, while the parameters in the low band (the range in this band is in a low end) may be updated within smaller dynamic range.

[0148] In some examples, parameters in one or more bands may be skipped. For example, the parameters in some categories may be fixed during the online training. For example, to reduce the weight update signaling, only the bias terms of a neural layer are updated, while the weights are not. Alternatively, only the weights or biases with larger value may be updated during the online training.

[0149] Furthermore, in some examples, a band may be further divided into multiple sub-bands including a first sub-band and a second sub-band. Parameters in the first sub-band may be updated while parameters in the second sub-band may be not updated, where the parameters in the first sub-band may have higher values than the parameters in the second sub-band. For examples, to reduce the weight update signaling, only the bias terms of a neural layer are updated, while the weights are not. Alternatively, only the weights or biases with larger value may be updated during the online training.

[0150] In some examples, one or more parameters of a subset of channels may be updated. The subset of channels may include channels that are part of a plurality of channels included in a neural network layer of the plurality of neural network layers.

[0151] In some examples, a penalty term may be generated based on a difference of a first parameter of the at least one neural network layer in the offline trained network between before and after an online training epoch and an updated loss function may be obtained by adding the penalty term to an existing loss function. In some examples, if the trainable parameters/weights are from the offline trained neural layers, the difference of the parameters/weights between and after each round of online training epoch is used to generate a penalty term, which can be added on the existing loss function to encourage smaller difference after update.

[0152] In some examples, a penalty term may be generated based on a value of a second parameter of the at least one neural network layer in the simple neural network connected to the offline trained network and an updated loss function may be obtained by adding the penalty term to an existing loss function. For examples, if the trainable parameters/weights are from the newly added neural layers, the value of the parameters/weights is used to generate a penalty term, which can be added on the existing loss function to encourage smaller parameters/weights.

[0153] In some examples, post-processing based constraint may be applied on the plurality of parameters after the online training stage. That is, after the online training, post-processing based constraint may be applied to the plurality of parameters of the online trained neural network before transmitting the plurality of parameters to a decoder.

[0154] In some examples, post-processing based constraint may be implemented by value approximating. For example, an update difference may be approximated to a predefined value, where the update difference may be a difference between a value of at least one parameter of the at least one neural network layer in the offline trained network and the at least one parameter may be updated according to the update difference. For example, the predefined value may be the power of 2.

[0155] In some examples, the update difference may be set to be 0 in response to determining that the update difference is smaller than a predefined threshold. For examples, when the update difference is below a threshold, the update difference may be set to be 0 such that the corresponding parameter remains the same as before the online training.

[0156] In some examples, a parameter of the at least one neural network layer in the simple neural network connected to the offline trained network may be approximated to a predefined value. For example, for the newly trained parameters/weights, which are from a new network only trained during online process, the value of each parameter/weight may be approximated to a specific value, such as to the power of 2.

[0157] In some examples, the value of the parameter may be set to be 0 in response to determining that value of the parameter is smaller than a predefined threshold. For examples, when the value of the parameter is below a threshold, the value of the parameter may be set to be 0.

[0158] In some other examples, there is provided a non-transitory computer readable storage medium 1004, having instructions stored therein. When the instructions are executed by one or more processors 1020, the instructions cause the processor to perform any method as described in FIG. 11 and above.

[0159] The description of the present disclosure has been presented for purposes of illustration and is not intended to be exhaustive or limited to the present disclosure. Many modifications, variations, and alternative implementations will be apparent to those of ordinary skill in the art having the benefit of the teachings presented in the foregoing descriptions and the associated drawings.

[0160] The examples were chosen and described in order to explain the principles of the disclosure, and to enable others skilled in the art to understand the disclosure for various implementations and to best utilize the underlying principles and various implementations with various modifications as are suited to the particular use contemplated. Therefore, it is to be understood that the scope of the disclosure is not to be limited to the specific examples of the implementations disclosed and that modifications and other implementations are intended to be included within the scope of the present disclosure

WHAT IS CLAIMED IS:

1. A method for hybrid training of neural networks for video coding, comprising:

obtaining, in an offline training stage, an offline trained network by training a neural network offline; and

refining, in an online training stage, a plurality of neural network layers with constraint on a plurality of parameters of the plurality of neural network layers, wherein the plurality of neural network layers comprise at least one neural network layer in the offline trained network or in a simple neural network connected to the offline trained network.

2. The method of claim 1, further comprising:

applying regularization based constraint on the plurality of parameters.

3. The method of claim 2, further comprising:

updating the plurality of parameters using a dynamic range before and after an online training epoch.

4. The method of claim 3, further comprising:

clipping a first parameter of the at least one neural network layer in the offline trained network using the dynamic range based on a value of the first parameter before the online training epoch and a predetermined value.

5. The method of claim 4, further comprising:

clipping a second parameter of the at least one neural network layer in the simple neural network connected to the offline trained network by a predefined range.

6. The method of claim 2, further comprising:

updating the plurality of parameters according to a parameter type of at least one parameter of the plurality of neural network layers.

7. The method of claim 6, further comprising:

classifying the plurality of parameters into different bands; and

updating one or more parameters in the different bands differently.

8. The method of claim 7, further comprising:

classifying the plurality of parameters based on one of followings: intensity values of the parameters, or roles of neurons.

9. The method of claim 7, wherein the different bands comprise a first band and a second band, wherein one or more parameters in the first band are updated using a first dynamic range and one or more parameters in the second band are updated using a second dynamic range, and wherein the first dynamic range is greater than the second dynamic range.

10. The method of claim 7, further comprising:

skipping updating one or more parameters in one or more bands.

11. The method of claim 7, further comprising:

dividing a band into a first sub-band and a second sub-band; and

updating one or more parameters in the first sub-band without updating one or more parameters in the second sub-band, wherein the one or more parameters in the first sub-band have higher values than the one or more parameters in the second sub-band.

12. The method of claim 2, further comprising:

updating one or more parameters of a subset of channels, wherein a neural network layer of the plurality of neural network layers comprises a plurality of channels comprising the subset of channels.

13. The method of claim 2, further comprising:

updating the plurality of parameters based on regularization of a loss function.

14. The method of claim 13, further comprising:

generating a penalty term based on a difference of a first parameter of the at least one neural network layer in the offline trained network between before and after an online training epoch; and

obtaining an updated loss function by adding the penalty term to an existing loss function.

15. The method of claim 13, further comprising:

generating a penalty term based on a value of a second parameter of the at least one neural network layer in the simple neural network connected to the offline trained network; and

obtaining an updated loss function by adding the penalty term to an existing loss function.

16. The method of claim 1, further comprising:

applying post-processing based constraint on the plurality of parameters after the online training stage.

17. The method of claim 16, further comprising:

approximating an update difference to a predefined value, wherein the update difference is a difference between a value of at least one parameter of the at least one neural network layer in the offline trained network; and

updating the at least one parameter according to the updated difference.

18. The method of claim 17, further comprising:

approximating the update difference by quantizing the update difference.

19. The method of claim 17, further comprising:

in response to determining that the update difference is smaller than a predefined threshold, determining that the update difference is 0.

20. The method of claim 16, further comprising:

approximating a parameter of the at least one neural network layer in the simple neural network connected to the offline trained network to a predefined value.

21. The method of claim 20, further comprising:

approximating the parameter by quantizing the parameter.

22. The method of claim 20, further comprising:

in response to determining that a value of the parameter is smaller than a predefined threshold, determining that the value of the parameter is 0.

23. An apparatus for hybrid training of neural networks for video coding, comprising:

one or more processors; and

a memory configured to store instructions executable by the one or more processors,

wherein the one or more processors, upon execution of the instructions, are configured to perform the method in any one of claims 1-22.

24. A non-transitory computer-readable storage medium storing computer-executable instructions that, when executed by one or more computer processors, causing the one or more computer processors to perform the method in any one of claims 1-22.

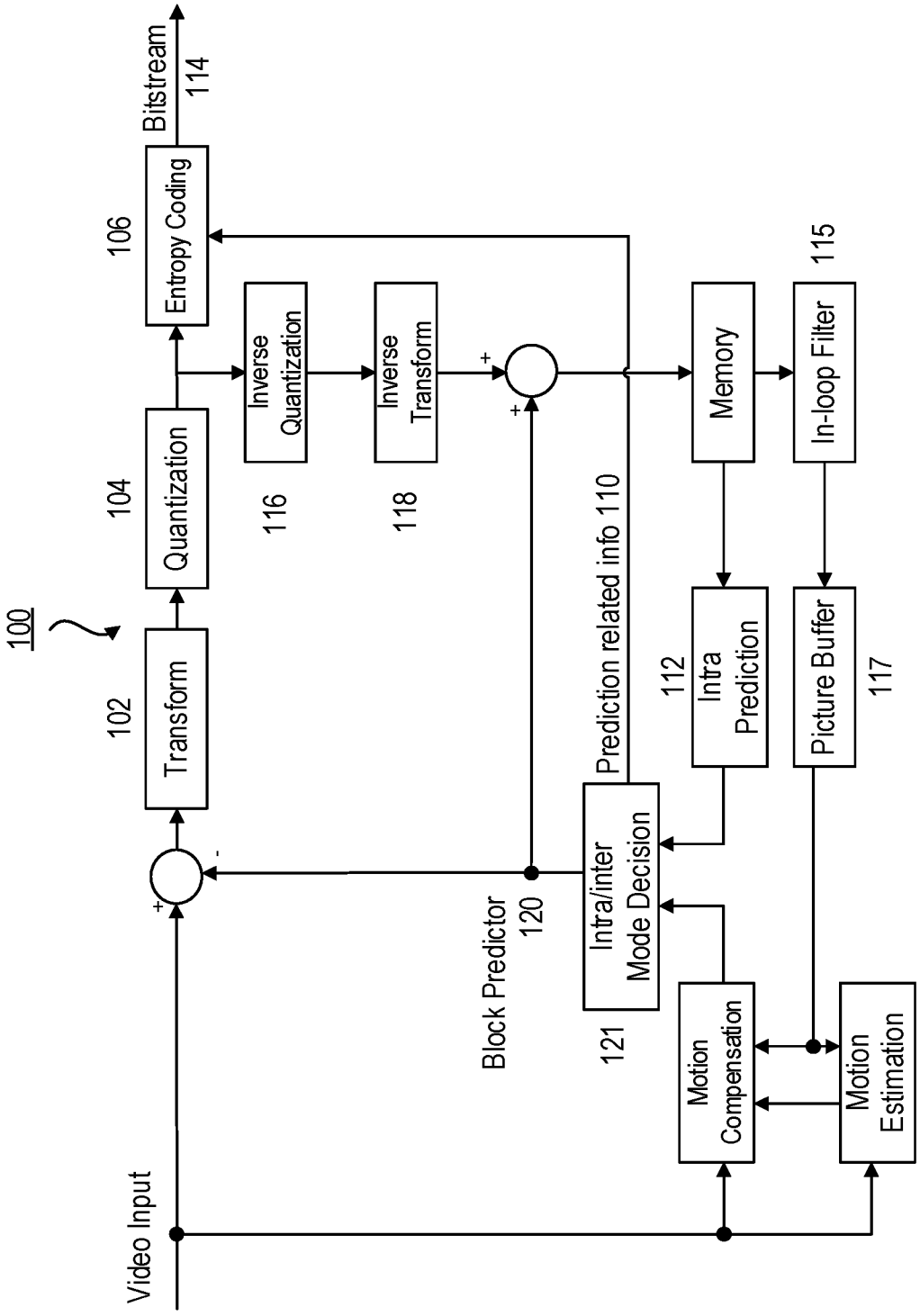

100
Video Input
Transform
102
Quantization
104
Entropy Coding
106
Bitstream
114
Inverse Quantization
116
Inverse Transform
118
Block Predictor
120
Intra/inter Mode Decision
121
Prediction related info 110
Intra Prediction
112
Memory
In-loop Filter
115
Picture Buffer
117
Motion Compensation
Motion Estimation

FIG. 1

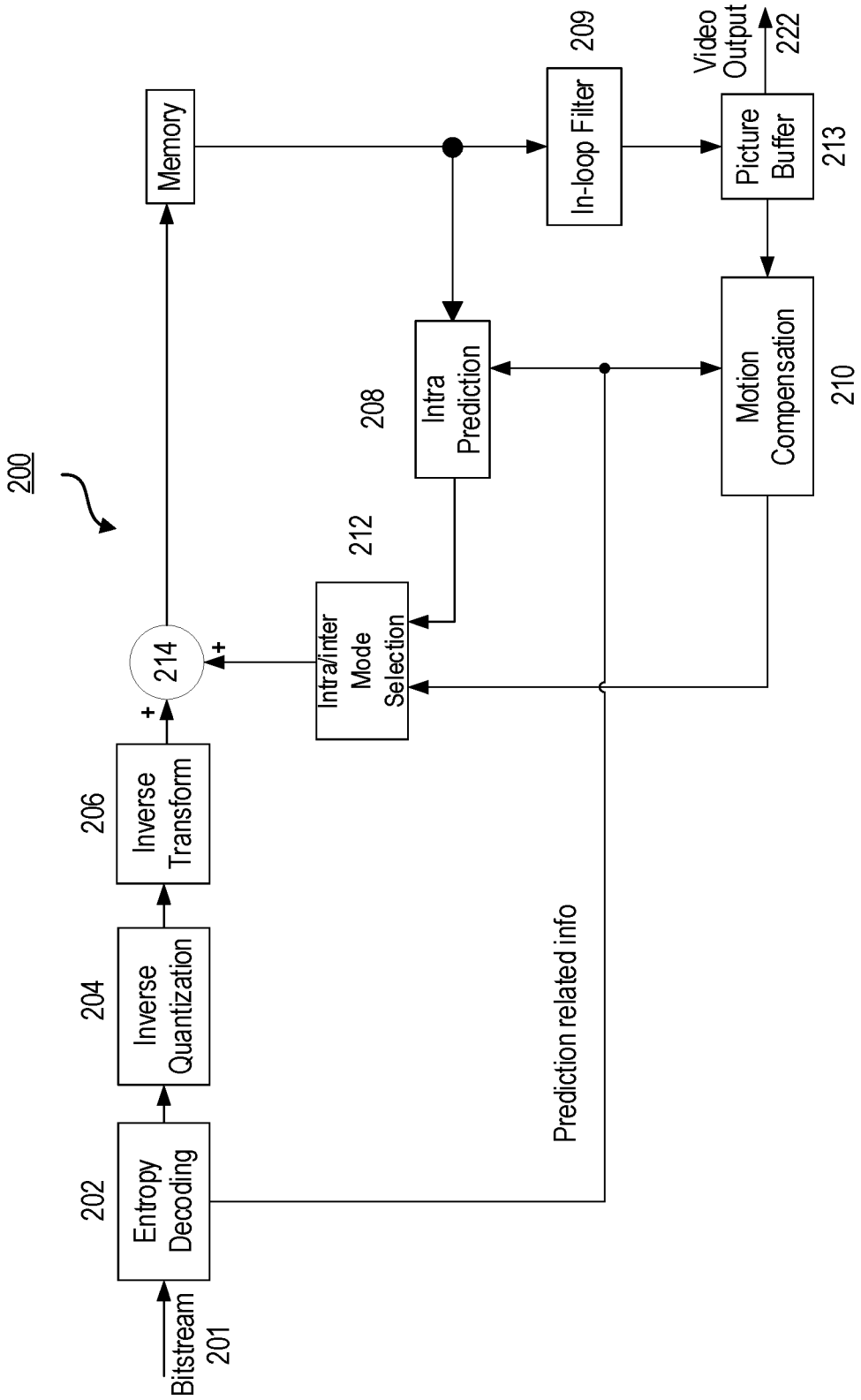

200
Bitstream
201
202
Entropy Decoding
204
Inverse Quantization
206
Inverse Transform
214
+
+
Intra/inter Mode Selection
212
208
Intra Prediction
Memory
209
In-loop Filter
Video Output
222
Picture Buffer
213
Motion Compensation
210
Prediction related info

FIG. 2

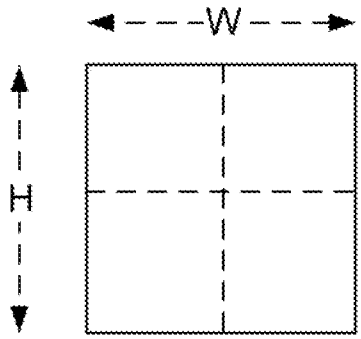


FIG. 3A

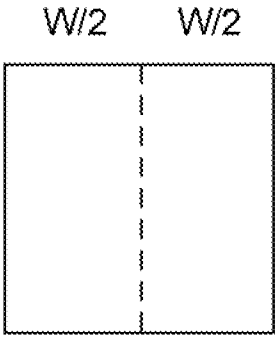


FIG. 3B

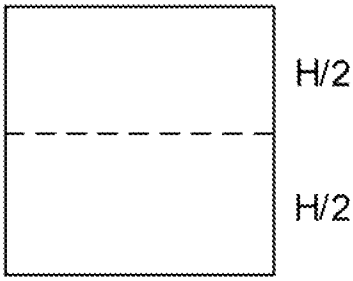


FIG. 3C

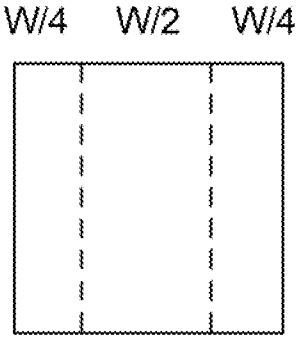


FIG. 3D

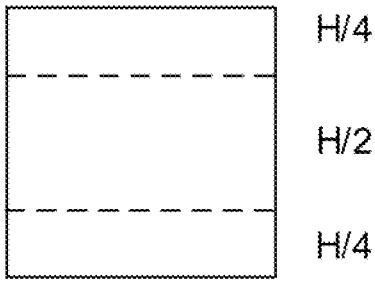


FIG. 3E

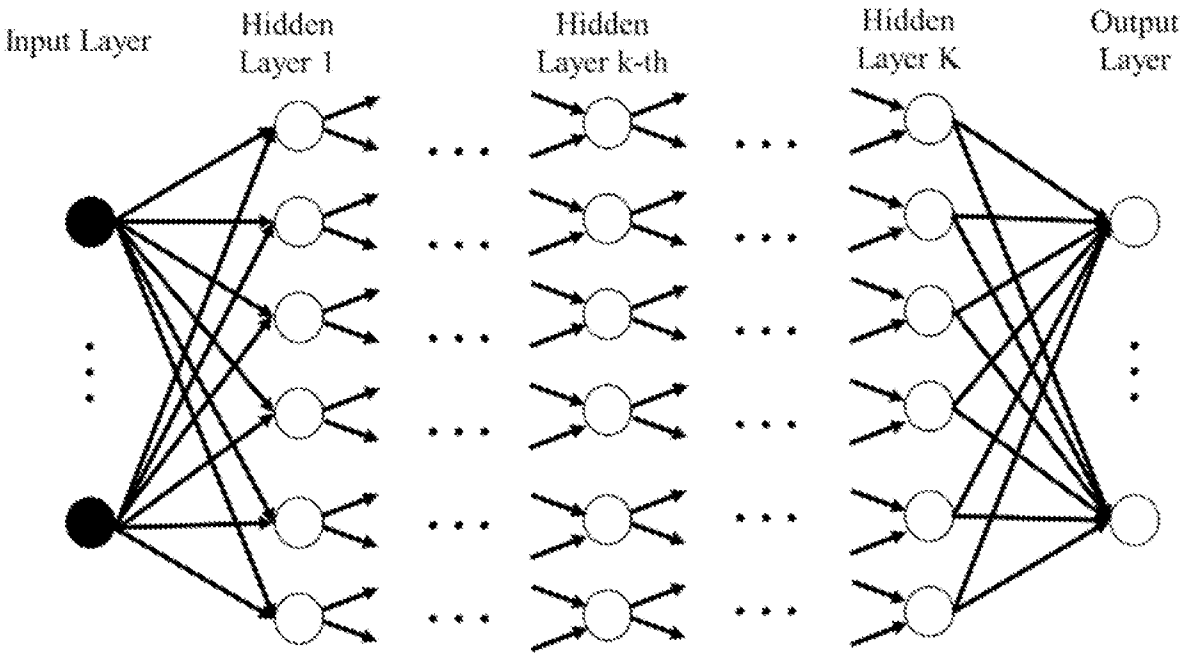

Input Layer
Hidden Layer 1
Hidden Layer k-th
Hidden Layer K
Output Layer

FIG. 4

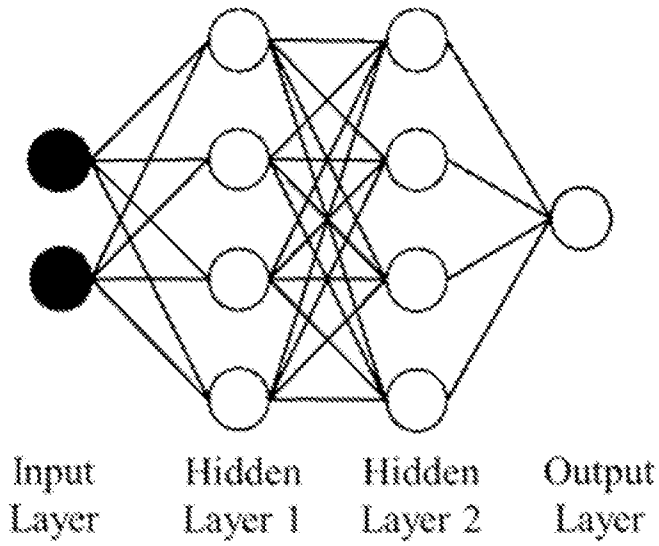

Input Layer
Hidden Layer 1
Hidden Layer 2
Output Layer

FIG. 5A

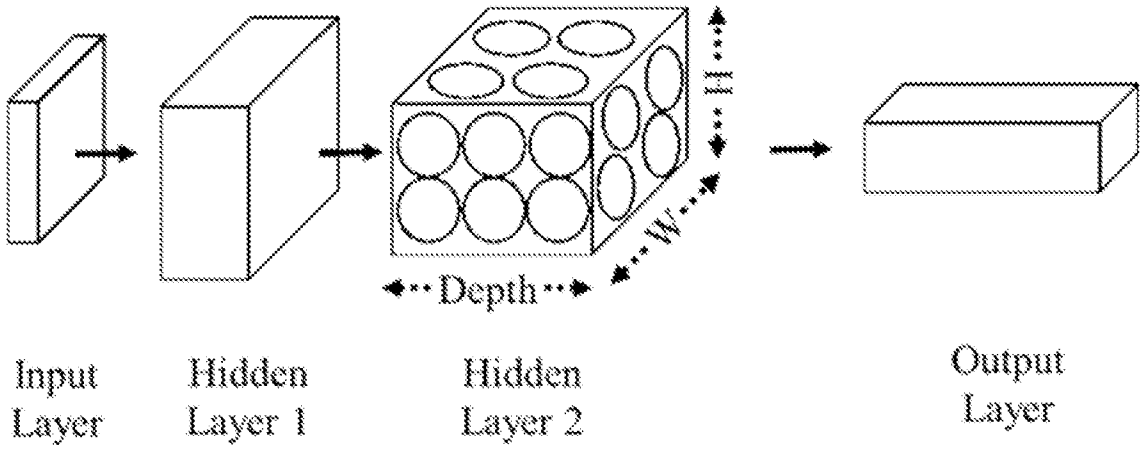

H
W
Depth
Input Layer
Hidden Layer 1
Hidden Layer 2
Output Layer

FIG. 5B

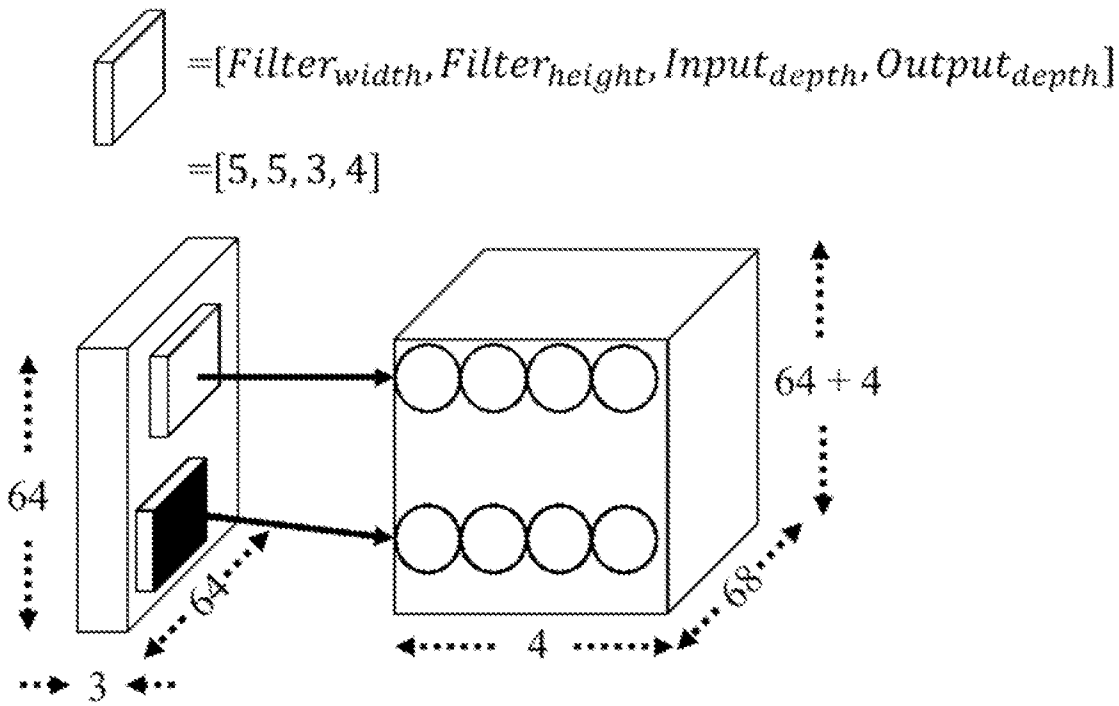

$=[Filter_{width}, Filter_{height}, Input_{depth}, Output_{depth}]$
$=[5, 5, 3, 4]$
64
64
3
4
64 + 4
68

FIG. 6

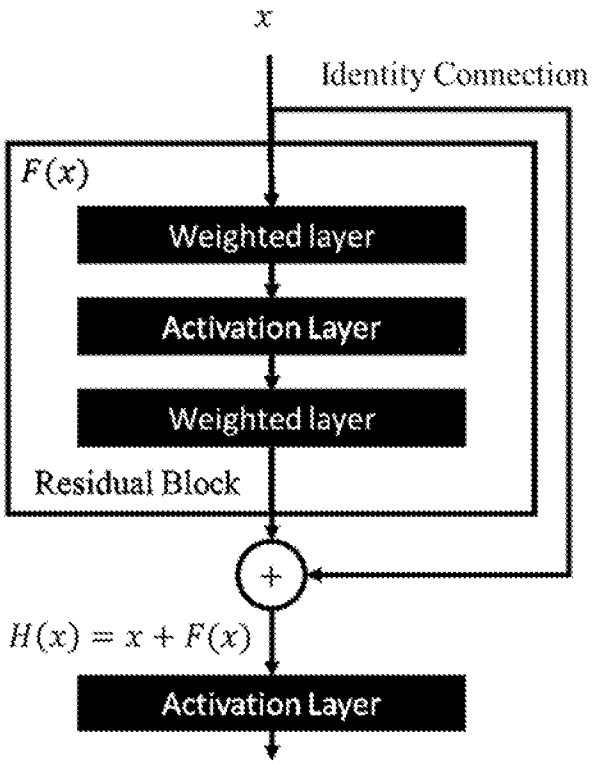

x
Identity Connection
F(x)
Weighted layer
Activation Layer
Weighted layer
Residual Block
+
H(x) = x + F(x)
Activation Layer

FIG. 7A

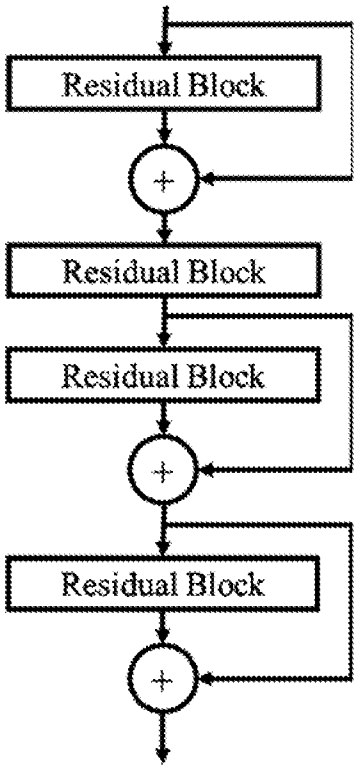

Residual Block
+
Residual Block
Residual Block
+
Residual Block
+

FIG. 7B

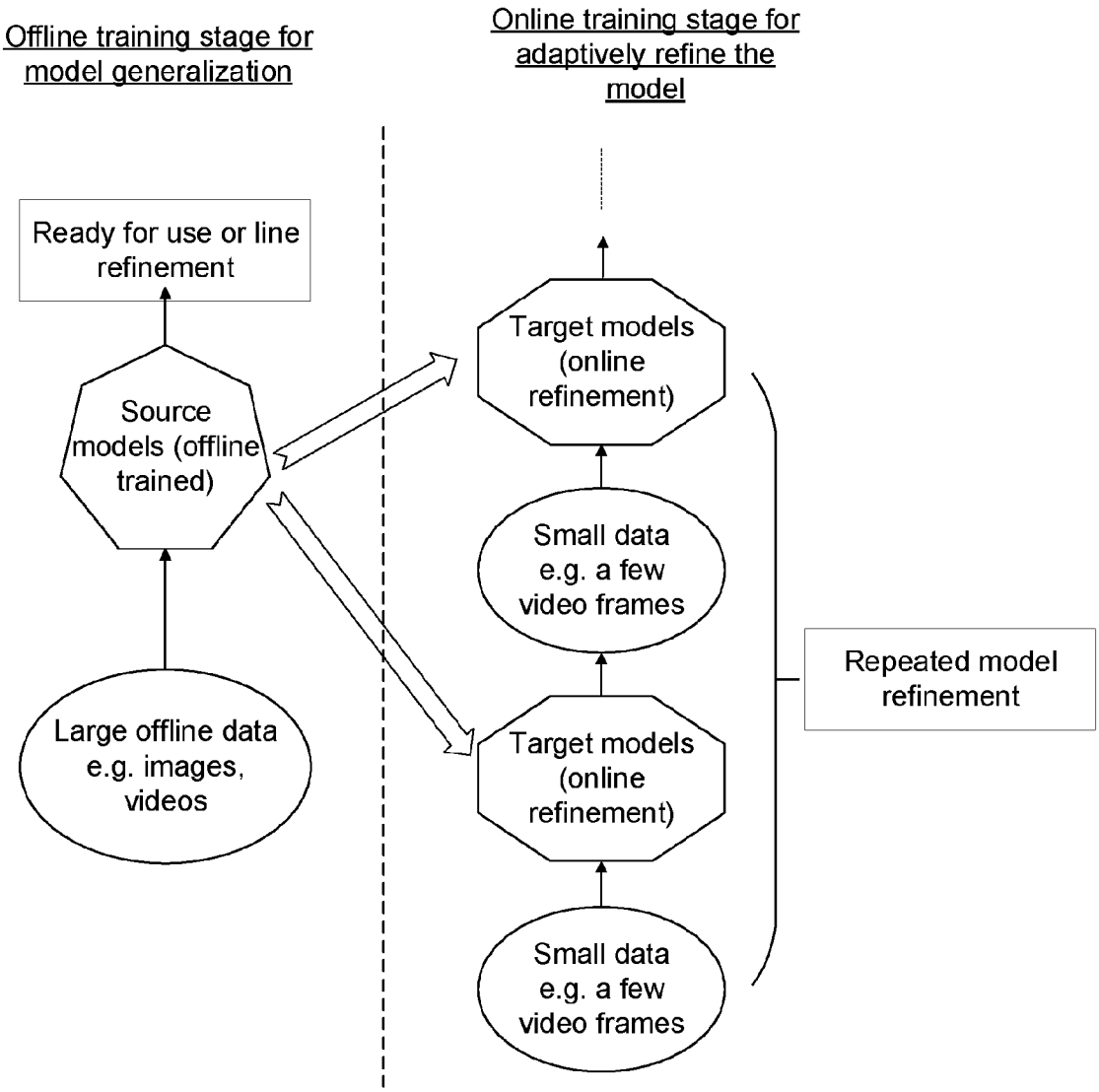

Offline training stage for model generalization
Online training stage for adaptively refine the model
Ready for use or line refinement
Source models (offline trained)
Large offline data e.g. images, videos
Target models (online refinement)
Small data e.g. a few video frames
Target models (online refinement)
Small data e.g. a few video frames
Repeated model refinement

FIG. 8

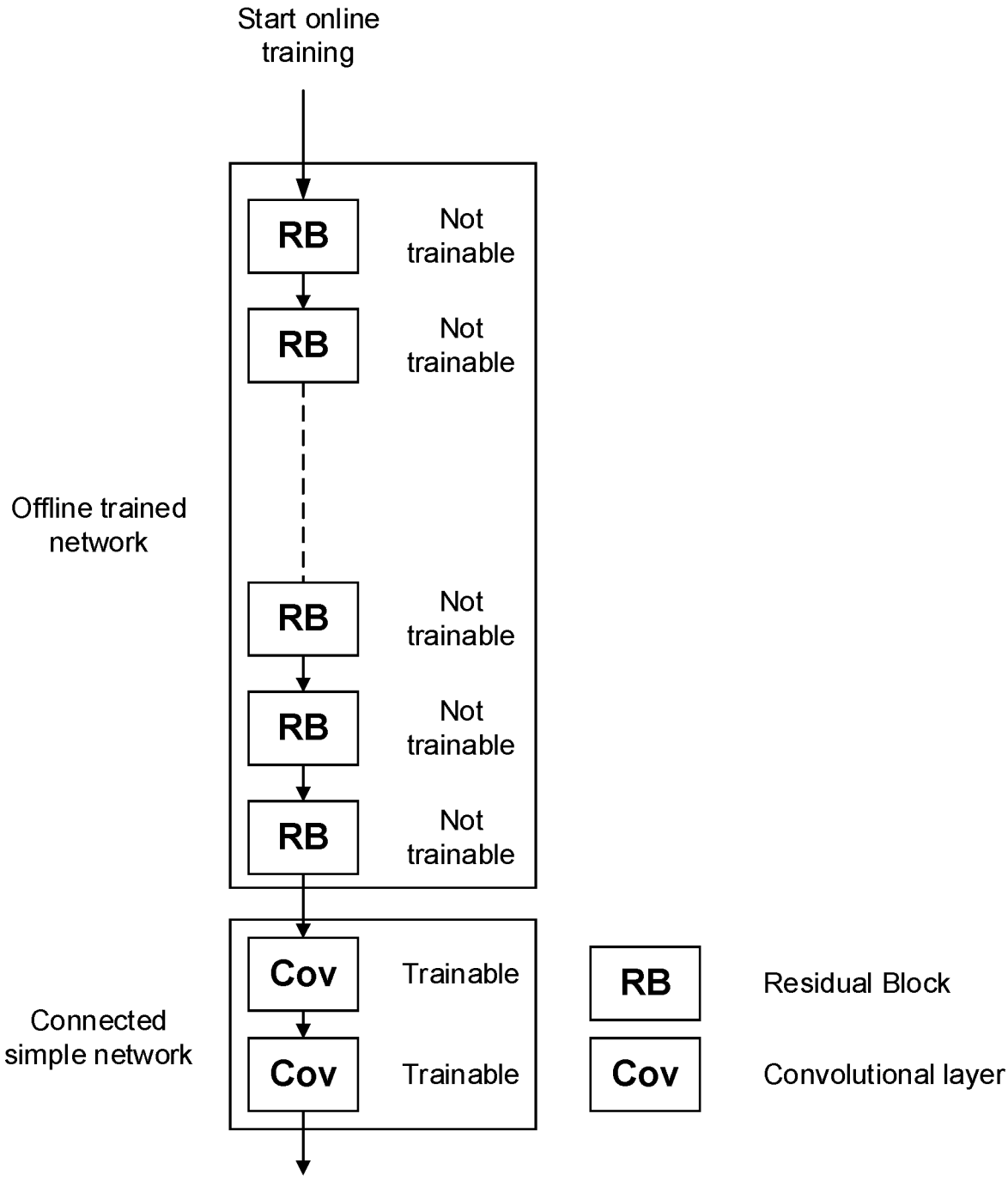

Start online training
RB
Not trainable
RB
Not trainable
Offline trained network
RB
Not trainable
RB
Not trainable
RB
Not trainable
Cov
Trainable
Connected simple network
Cov
Trainable
RB
Residual Block
Cov
Convolutional layer

FIG. 9A

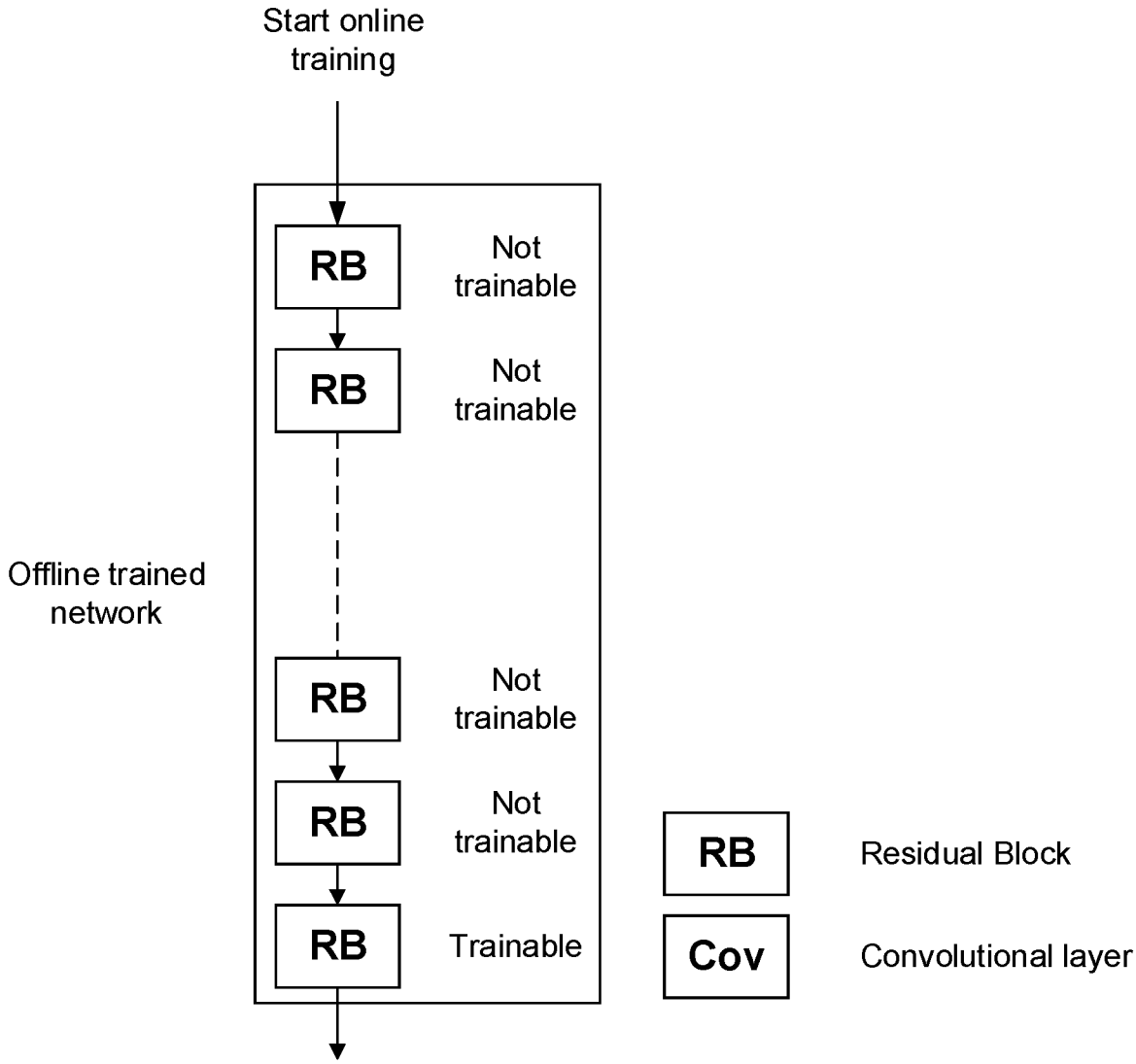

Start online training
RB
Not trainable
RB
Not trainable
Offline trained network
RB
Not trainable
RB
Not trainable
RB
Trainable
RB
Residual Block
Cov
Convolutional layer

FIG. 9B

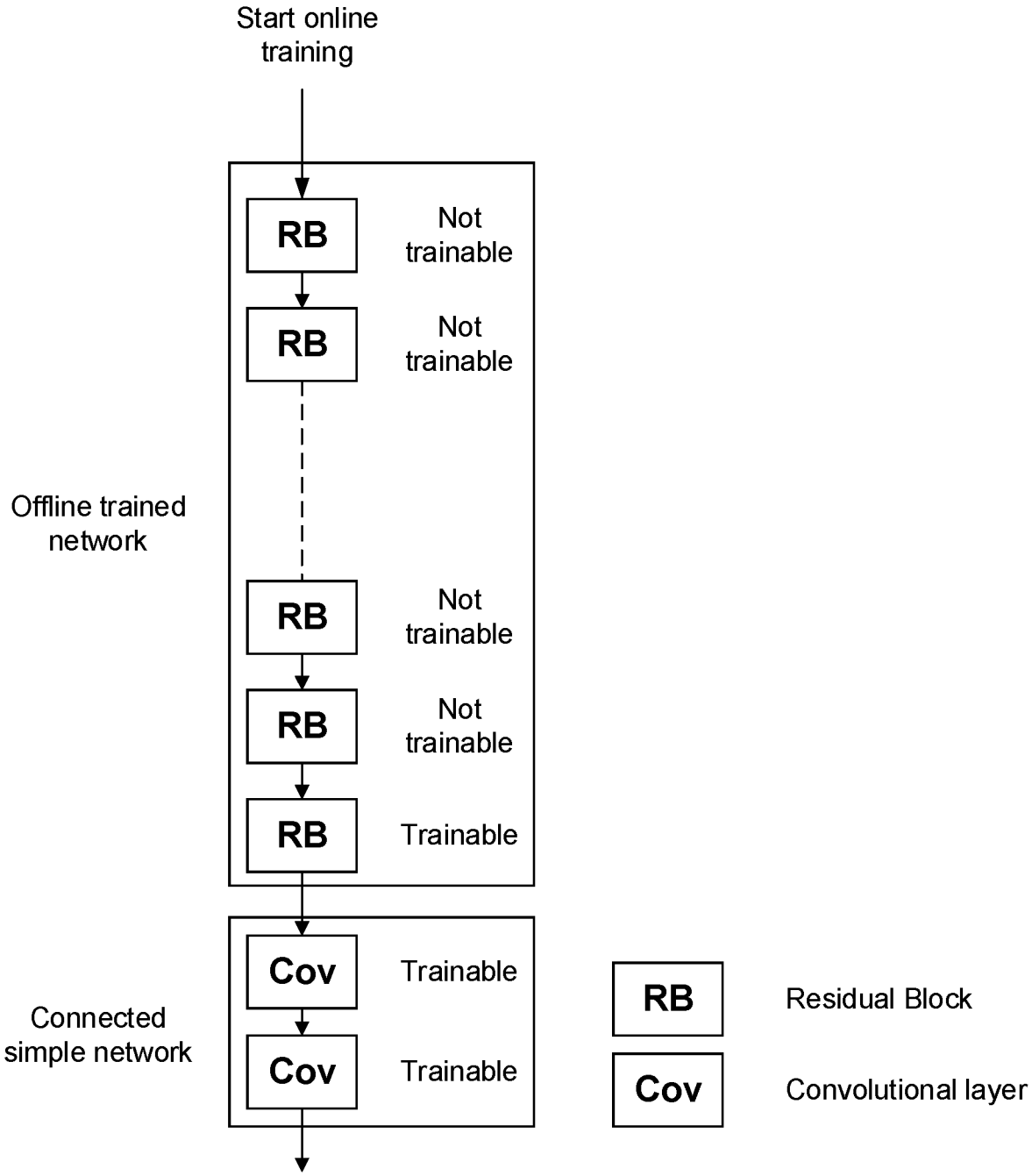

Start online training
RB
Not trainable
RB
Not trainable
Offline trained network
RB
Not trainable
RB
Not trainable
RB
Trainable
Cov
Trainable
Connected simple network
Cov
Trainable
RB
Residual Block
Cov
Convolutional layer

FIG. 9C

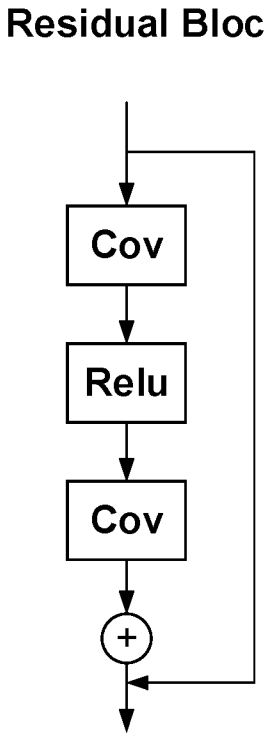


FIG. 9D

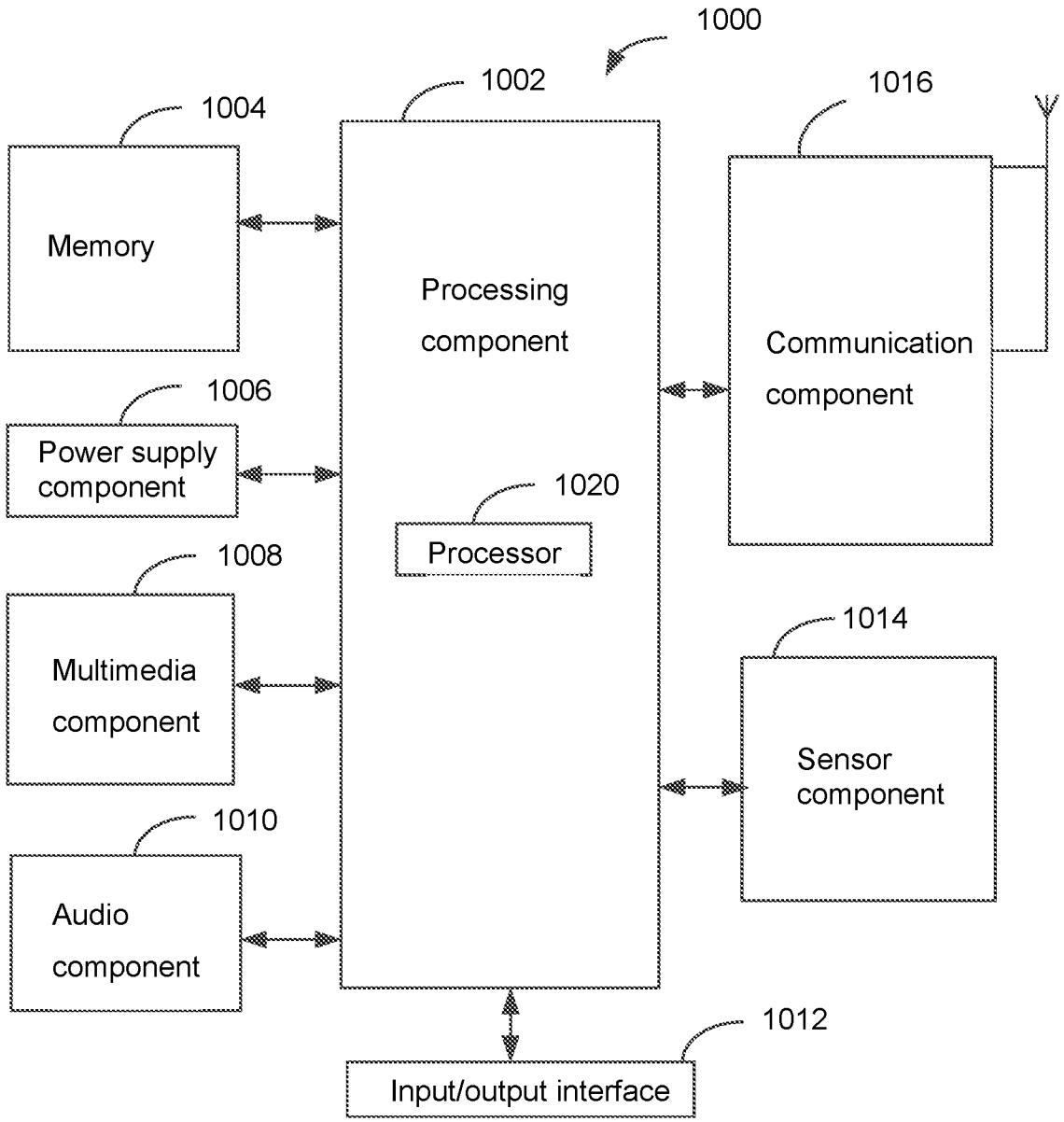

1000
1002
1004
1016
Memory
Processing component
Communication component
1006
Power supply component
1020
Processor
1008
Multimedia component
1014
Sensor component
1010
Audio component
1012
Input/output interface

FIG. 10

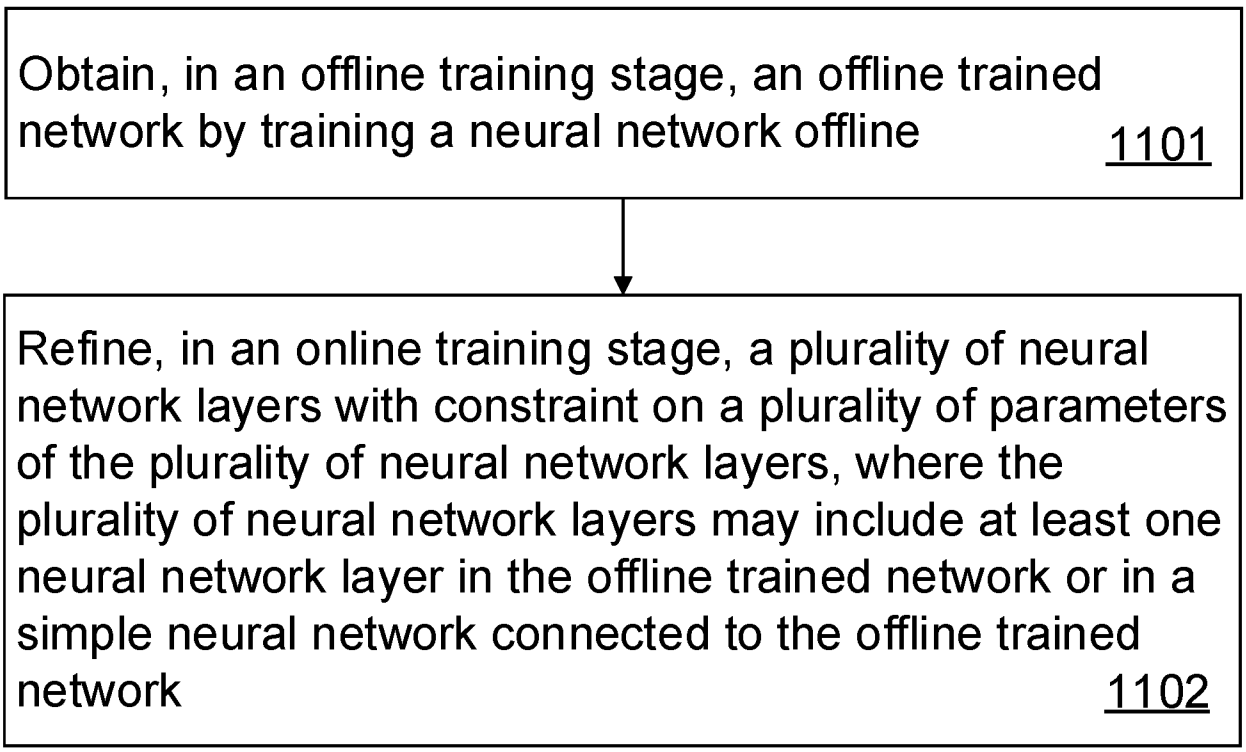

Obtain, in an offline training stage, an offline trained network by training a neural network offline
1101
Refine, in an online training stage, a plurality of neural network layers with constraint on a plurality of parameters of the plurality of neural network layers, where the plurality of neural network layers may include at least one neural network layer in the offline trained network or in a simple neural network connected to the offline trained network
1102

FIG. 11

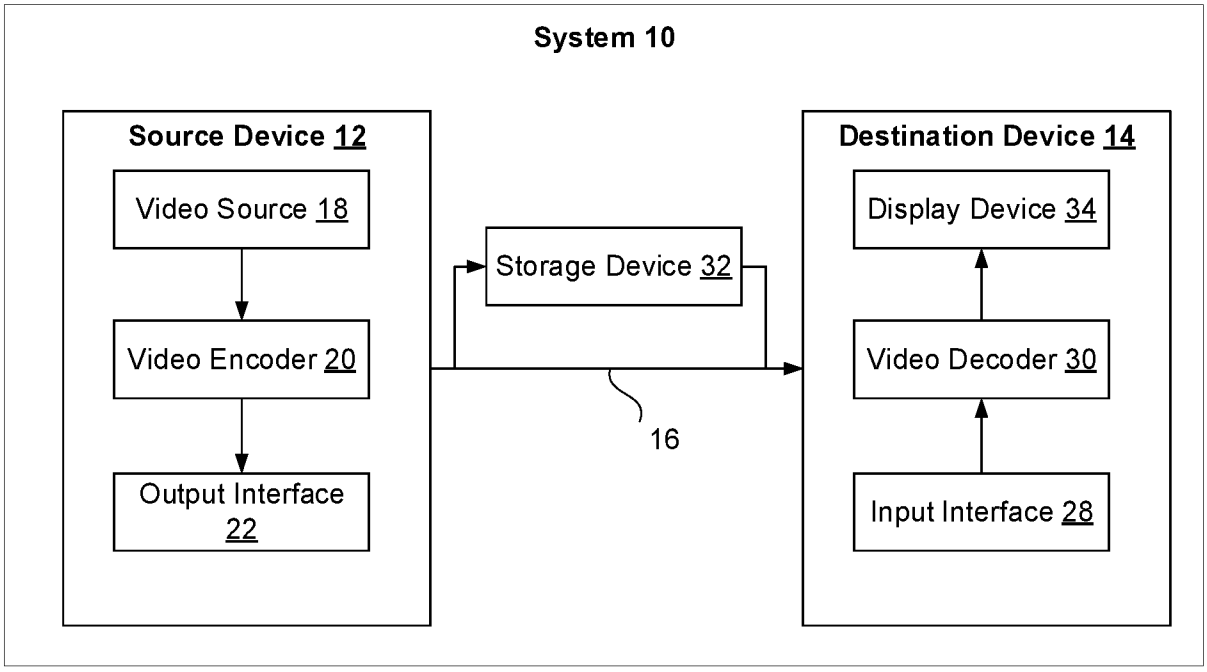

System 10
Source Device 12
Video Source 18
Video Encoder 20
Output Interface 22
Storage Device 32
16
Destination Device 14
Display Device 34
Video Decoder 30
Input Interface 28

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2022/038711

A. CLASSIFICATION OF SUBJECT MATTER

G06T 9/00(2006.01)i; **H04N 19/117**(2014.01)i; **G06N 3/04**(2006.01)i; **G06N 3/08**(2006.01)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)

G06T 9/00(2006.01); G06F 30/27(2020.01); G06N 3/04(2006.01); G06N 3/08(2006.01); G06Q 30/02(2012.01); G06Q 50/20(2012.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: offline, online, training, network, parameter

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 112417765 A (EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY) 26 February 2021 (2021-02-26) claims 1-6	1-24
Y	US 2016-0217369 A1 (QUALCOMM INCORPORATED) 28 July 2016 (2016-07-28) paragraphs [0065]-[0077] and claim 1	1-24
A	US 2019-0370638 A1 (THALES CANADA INC.) 05 December 2019 (2019-12-05) claims 1-6	1-24
A	EP 3846109 A2 (BEIJING BAIDU NETCOM SCIENCE AND TECHNOLOGY CO. LTD.) 07 July 2021 (2021-07-07) claims 1-9	1-24
A	US 2019-0188568 A1 (SALESFORCE.COM, INC.) 20 June 2019 (2019-06-20) paragraphs [0020]-[0050]	1-24

☐ 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	Date of mailing of the international search report
15 November 2022	**15 November 2022**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office **189 Cheongsa-ro, Seo-gu, Daejeon** **35208, Republic of Korea**	**YANG, Jeong Rok**
Facsimile No. **+82-42-481-8578**	Telephone No. **+82-42-481-5709**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2022/038711

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	112417765	A	26 February 2021	None			
US	2016-0217369	A1	28 July 2016	BR	112017015560	A2	13 March 2018
				CN	107004157	A	01 August 2017
				EP	3248148	A1	29 November 2017
				JP	2018-506785	A	08 March 2018
				KR	10-2017-0106338	A	20 September 2017
				TW	201627923	A	01 August 2016
				US	10223635	B2	05 March 2019
				WO	2016-118257	A1	28 July 2016
US	2019-0370638	A1	05 December 2019	CA	3101593	A1	05 December 2019
				EP	3803703	A1	14 April 2021
				KR	10-2021-0015914	A	10 February 2021
				SG	11202011621	A	30 December 2020
				WO	2019-229728	A1	05 December 2019
EP	3846109	A2	07 July 2021	CN	112070226	A	11 December 2020
				EP	3846109	A3	17 November 2021
				JP	2022-023776	A	08 February 2022
				KR	10-2021-0052410	A	10 May 2021
				US	2021-0248513	A1	12 August 2021
US	2019-0188568	A1	20 June 2019	US	11276002	B2	15 March 2022