



(51) International Patent Classification:

H04N 19/117 (2014.01) *G06N 3/08* (2023.01)
H04N 19/85 (2014.01) *G06V 10/20* (2022.01)
G06N 3/04 (2023.01) *G06T 9/00* (2006.01)

(21) International Application Number:

PCT/FI2022/050663

(22) International Filing Date:

05 October 2022 (05.10.2022)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

20216121 29 October 2021 (29.10.2021) FI

(71) Applicant: **NOKIA TECHNOLOGIES OY** [FI/FI];
Karakaari 7, 02610 Espoo (FI).

(72) Inventors: **ZHANG, Honglei**; Kukkaniitynkatu 4 B, 33710
Tampere (FI). **CRICRI, Francesco**; Alaverstaanraitti 6 A
8, 33100 Tampere (FI). **HANNUKSELA, Miska, Ma-**
tias; Rusthollinrinne 2, 33610 Tampere (FI). **REZAZADE-**

GAN TAVAKOLI, Hamed; Haukilahdenranta 21 A 2,
02170 Espoo (FI). **LE, Nam, Hai**; Annalankatu 11 C
29, 33710 Tampere (FI). **GHAZNAVI YOUVALARI,**
Ramin; Myrskynkatu 12 A 32, 33900 Tampere (FI). **AHO-**
NEN, Jukka, Ilari; Kalevan puistotie 13 B 45, 33500 Tam-
pere (FI). **AKSU, Emre, Baris**; Virrenkatu 3 D 20, 33800
Tampere (FI).

(74) Agent: **NOKIA TECHNOLOGIES OY** et al.; Mikko
Alatossava, IPR Department, Karakaari 7, 02610 Espoo
(FI).

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE,
KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU,
LY, MA, MD, MG, MK, MN, 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,

(54) Title: A METHOD, AN APPARATUS AND A COMPUTER PROGRAM PRODUCT FOR VIDEO CODING

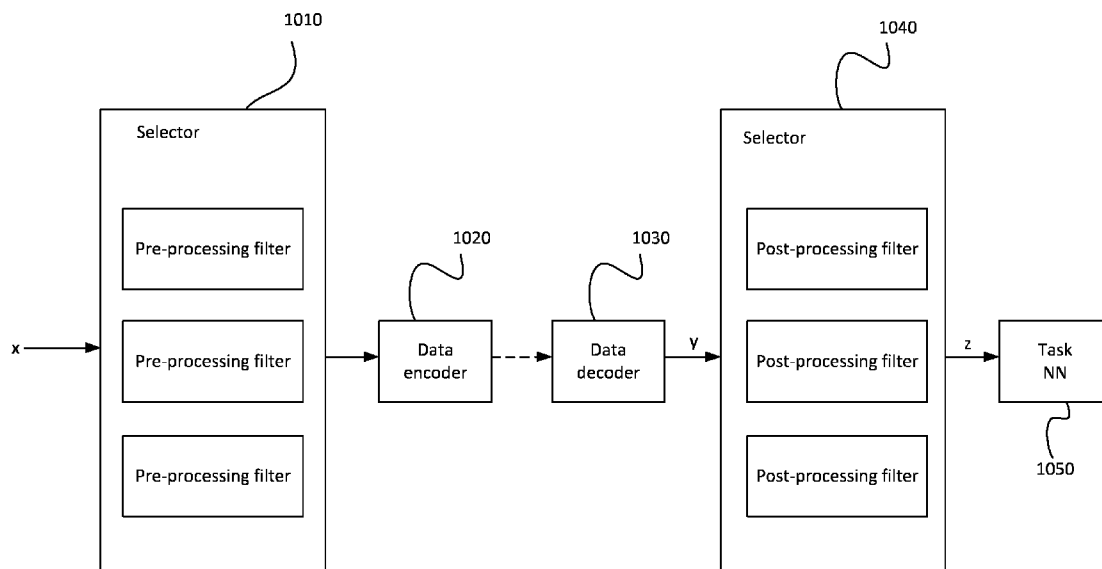


Fig. 10

(57) Abstract: The embodiments relate to a method for encoding and decoding, and a technical equipment for implementing the methods. The method for encoding comprises receiving input media data; preprocessing the input media data by one or more pre-processing filters (1010); compressing the preprocessed input media data by a video codec (1020); and including information characterizing the one or more pre-processing filters (1010) in or along the compressed media data.

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, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

A METHOD, AN APPARATUS AND A COMPUTER PROGRAM PRODUCT FOR VIDEO CODING

5 The project leading to this application has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 783162. The JU receives support from the European Union's Horizon 2020 research and innovation program and Netherlands, Czech Republic, Finland, Spain, Italy.

10 The project leading to this application has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876019. The JU receives support from the European Union's Horizon 2020 research and innovation program and Germany, Netherlands, Austria, Romania, France, Sweden, Cyprus, Greece, Lithuania, Portugal, Italy, Finland, Turkey.

15 Technical Field

The present solution generally relates to video coding. In particular, the solution relates to video coding for machines (VCM).

20 Background

One of the elements in image and video compression is to compress data while maintaining the quality to satisfy human perceptual ability. However, in recent development of machine learning, machines can replace humans when
25 analyzing data for example in order to detect events and/or objects in video/image. Thus, when decoded image data is consumed by machines, the quality of the compression can be different from the human approved quality. Therefore a concept Video Coding for Machines (VCM) has been provided.

30 Summary

The scope of protection sought for various embodiments of the invention is set out by the independent claims. The embodiments and features, if any, described in this specification that do not fall under the scope of the independent claims are to be
35 interpreted as examples useful for understanding various embodiments of the invention.

Various aspects include a method, an apparatus and a computer readable medium comprising a computer program stored therein, which are characterized by what is stated in the independent claims. Various embodiments are disclosed in the dependent claims.

5

According to a first aspect, there is provided an apparatus for encoding comprising means for receiving input media data; means for pre-processing the input media data by one or more pre-processing filters; means for compressing the pre-processed input media data by a video codec; and means for including information characterizing the one or more pre-processing filters in or along the compressed media data.

10

15

According to a second aspect, there is provided an apparatus for decoding comprising means for receiving compressed media data; means for decompressing the compressed media data; means for decoding information characterizing one or more pre-processing filters from or along the compressed media data, the one or more pre-processing filters having been applied to input media data in generation of the compressed media data; means for selecting one or more post-processing filters on the basis of the information characterizing the one or more pre-processing filters; means for post-processing the decompressed input media data by one or more post-processing filters; and means for transferring the post-processed decompressed data for analysis at one or more task neural networks.

20

25

According to a third aspect, there is provided a method for encoding, comprising: receiving input media data; pre-processing the input media data by one or more pre-processing filters; compressing the pre-processed input media data by a video codec; and including information characterizing the one or more pre-processing filters in or along the compressed media data.

30

35

According to a fourth aspect, there is provided a method for decoding, comprising receiving compressed media data; decompressing the compressed media data; decoding information characterizing one or more pre-processing filters from or along the compressed media data, the one or more pre-processing filters having been applied to input media data in generation of the compressed media data; selecting one or more post-processing filters on the basis of the information characterizing the one or more pre-processing filters; post-processing the decompressed input

media data by one or more post-processing filters; and transferring the post-processed decompressed data for analysis at one or more task neural networks.

5 According to a fifth aspect, there is provided an apparatus comprising at least one processor, memory including computer program code, the memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the following: receive input media data; pre-process the input media data by one or more pre-processing filters; compress the pre-processed input media data by a video codec; and include information
10 characterizing the one or more pre-processing filters in or along the compressed media data.

According to a sixth aspect, there is provided an apparatus comprising at least one processor, memory including computer program code, the memory and the computer program code configured to, with the at least one processor, cause the
15 apparatus to perform at least the following: receive compressed media data; decompress the compressed media data; decode information characterizing one or more pre-processing filters from or along the compressed media data, the one or more pre-processing filters having been applied to input media data in generation
20 of the compressed media data; select one or more post-processing filters on the basis of the information characterizing the one or more pre-processing filters; post-process the decompressed input media data by one or more post-processing filters; and transfer the post-processed decompressed data for analysis at one or more task neural networks.

25 According to a seventh aspect, there is provided computer program product comprising computer program code configured to, when executed on at least one processor, cause an apparatus or a system to: receive input media data; pre-process the input media data by one or more pre-processing filters; compress the pre-processed input media data by a video codec; and include information
30 characterizing the one or more pre-processing filters in or along the compressed media data.

According to an eighth aspect, there is provided computer program product comprising computer program code configured to, when executed on at least one
35 processor, cause an apparatus or a system to: receive compressed media data; decompress the compressed media data; decode information characterizing one or

more pre-processing filters from or along the compressed media data, the one or more pre-processing filters having been applied to input media data in generation of the compressed media data; select one or more post-processing filters on the basis of the information characterizing the one or more pre-processing filters; post-process the decompressed input media data by one or more post-processing filters; and transfer the post-processed decompressed data for analysis at one or more task neural networks.

According to an embodiment, the pre-processing filter is configured to perform one or more of the following: smoothing areas that are unimportant for machine tasks; enhancing video quality for better machine performance; enhancing salient objects in the media data; obscuring or concealing certain parts of a video.

According to an embodiment, the pre-processing filter is content adaptive.

According to an embodiment, the information characterizing the one or more pre-processing filters comprises one or more of the following: a purpose of said one or more pre-processing filters; a type of said one or more pre-processing filters; a topology of the one or more pre-processing filters; values for parameters of said one or more pre-processing filters.

According to an embodiment, one or more selector modules are configured to select the optimal one or more pre-processing filters and one or more post-processing filters based on the content of the input media data and/or target machine tasks and/or transferred information.

According to an embodiment, said one or more pre-processing filters and said one or more post-processing filters are any combination of the following: linear or non-linear transformation filters; neural network based filters.

According to an embodiment, said one or more post-processing filters are adapted by minimizing a loss function.

According to an embodiment, the loss function is defined for a data input to one or more pre-processing filters and reconstructed data from one or more post-processing filters

According to an embodiment, the loss function is defined for a measurement of an input data and a measurement of an output from one or more post-processing filters.

5 According to an embodiment, the loss function is defined for a data input to one or more post-processing filters and data output from one or more pre-processing filters.

According to an embodiment, the computer program product is embodied on a non-transitory computer readable medium.

10 Description of the Drawings

In the following, various embodiments will be described in more detail with reference to the appended drawings, in which

15 Fig. 1 shows an example of a codec with neural network (NN) components;

Fig. 2 shows another example of a video coding system with neural network components;

20 Fig. 3 shows an example of a neural auto-encoder architecture;

Fig. 4 shows an example of a neural network-based end-to-end learned video coding system;

25 Fig. 5 shows an example of a video coding for machines;

Fig. 6 shows an example of a pipeline for end-to-end learned approach;

Fig. 7 shows an example of training an end-to-end learned system;

30

Fig. 8 shows an example of a system including an encoder, a decoder, a post-processing filter and a set of task-NNs;

Fig. 9 shows a VCM system according to an embodiment;

35

Fig. 10 shows a VCM system according to another embodiment;

Fig. 11 shows components, data and loss function at VCM decoder side for adaptation training according to an embodiment;

Fig. 12 shows components, data, calculation of measurements and loss function used for adaptation training according to an embodiment;

Fig. 13 shows components, data and loss function used in the adaptation training according to an embodiment;

Fig. 14 is a flowchart illustrating a method for encoding according to an embodiment;

Fig. 15 is a flowchart illustrating a method for decoding according to an embodiment; and

Fig. 16 shows an apparatus according to an embodiment.

Description of Example Embodiments

The following description and drawings are illustrative and are not to be construed as unnecessarily limiting. The specific details are provided for a thorough understanding of the disclosure. However, in certain instances, well-known or conventional details are not described in order to avoid obscuring the description. References to one or an embodiment in the present disclosure can be, but not necessarily are, reference to the same embodiment and such references mean at least one of the embodiments.

Reference in this specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the disclosure.

The present embodiments relate to video coding for machines and in particular to pre-processing filter-aware decoder for video coding for machines.

A neural network (NN) is a computation graph consisting of several layers of computation. Each layer consists of one or more units, where each unit performs an

elementary computation. A unit is connected to one or more other units, and the connection may have associated with a weight. The weight may be used for scaling the signal passing through the associated connection. Weights are learnable parameters, i.e., values which can be learned from training data. There may be other learnable parameters, such as those of batch-normalization layers.

Two of the most widely used architectures for neural networks are feed-forward and recurrent architectures. Feed-forward neural networks are such that there is no feedback loop: each layer takes input from one or more of the layers before and provides its output as the input for one or more of the subsequent layers. Also, units inside a certain layer take input from units in one or more of preceding layers, and provide output to one or more of following layers.

Initial layers (those close to the input data) extract semantically low-level features such as edges and textures in images, and intermediate and final layers extract more high-level features. After the feature extraction layers there may be one or more layers performing a certain task, such as classification, semantic segmentation, object detection, denoising, style transfer, super-resolution, etc. In recurrent neural nets, there is a feedback loop, so that the network becomes stateful, i.e., it is able to memorize information or a state.

Neural networks are being utilized in an ever-increasing number of applications for many different types of device, such as mobile phones. Examples include image and video analysis and processing, social media data analysis, device usage data analysis, etc.

One of the important properties of neural networks (and other machine learning tools) is that they are able to learn properties from input data, either in supervised way or in unsupervised way. Such learning is a result of a training algorithm, or of a meta-level neural network providing the training signal.

In general, the training algorithm consists of changing some properties of the neural network so that its output is as close as possible to a desired output. For example, in the case of classification of objects in images, the output of the neural network can be used to derive a class or category index which indicates the class or category that the object in the input image belongs to. Training usually happens by minimizing or decreasing the output's error, also referred to as the loss. Examples of losses are

mean squared error, cross-entropy, etc. In recent deep learning techniques, training is an iterative process, where at each iteration the algorithm modifies the weights of the neural net to make a gradual improvement of the network's output, i.e., to gradually decrease the loss.

5

In this description, terms "model" and "neural network" are used interchangeably, and also the weights of neural networks are sometimes referred to as learnable parameters or simply as parameters.

10

Training a neural network is an optimization process, but the final goal is different from the typical goal of optimization. In optimization, the only goal is to minimize a function. In machine learning, the goal of the optimization or training process is to make the model learn the properties of the data distribution from a limited training dataset. In other words, the goal is to learn to use a limited training dataset in order to learn to generalize to previously unseen data, i.e., data which was not used for training the model. This is usually referred to as generalization. In practice, data may be split into at least two sets, the training set and the validation set. The training set is used for training the network, i.e., to modify its learnable parameters in order to minimize the loss. The validation set is used for checking the performance of the network on data which was not used to minimize the loss, as an indication of the final performance of the model. In particular, the errors on the training set and on the validation set are monitored during the training process to understand the following things:

15

20

25

30

- If the network is learning at all – in this case, the training set error should decrease, otherwise the model is in the regime of underfitting.
- If the network is learning to generalize – in this case, also the validation set error needs to decrease and to be not too much higher than the training set error. If the training set error is low, but the validation set error is much higher than the training set error, or it does not decrease, or it even increases, the model is in the regime of overfitting. This means that the model has just memorized the training set's properties and performs well only on that set, but performs poorly on a set not used for tuning its parameters.

35

Lately, neural networks have been used for compressing and de-compressing data such as images, i.e., in an image codec. The most widely used architecture for realizing one component of an image codec is the auto-encoder, which is a neural network consisting of two parts: a neural encoder and a neural decoder. The neural

encoder takes as input an image and produces a code which requires less bits than the input image. This code may be obtained by applying a binarization or quantization process to the output of the encoder. The neural decoder takes in this code and reconstructs the image which was input to the neural encoder.

5

Such neural encoder and neural decoder may be trained to minimize a combination of bitrate and distortion, where the distortion may be based on one or more of the following metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), or similar. These distortion metrics are meant to be correlated to the human visual perception quality, so that minimizing or maximizing one or more of these distortion metrics results into improving the visual quality of the decoded image as perceived by humans.

10

Video codec comprises an encoder that transforms the input video into a compressed representation suited for storage/transmission and a decoder that can decompress the compressed video representation back into a viewable form. An encoder may discard some information in the original video sequence in order to represent the video in a more compact form (that is, at lower bitrate).

15

Hybrid video codecs, for example ITU-T H.263 and H.264, may encode the video information in two phases. Firstly pixel values in a certain picture area (or "block") are predicted for example by motion compensation means (finding and indicating an area in one of the previously coded video frames that corresponds closely to the block being coded) or by spatial means (using the pixel values around the block to be coded in a specified manner). Secondly the prediction error, i.e. the difference between the predicted block of pixels and the original block of pixels, is coded. This may be done by transforming the difference in pixel values using a specified transform (e.g. Discrete Cosine Transform (DCT) or a variant of it), quantizing the coefficients and entropy coding the quantized coefficients. By varying the fidelity of the quantization process, encoder can control the balance between the accuracy of the pixel representation (picture quality) and size of the resulting coded video representation (file size or transmission bitrate).

20

25

30

Inter prediction, which may also be referred to as temporal prediction, motion compensation, or motion-compensated prediction, exploits temporal redundancy. In inter prediction the sources of prediction are previously decoded pictures.

35

Intra prediction utilizes the fact that adjacent pixels within the same picture are likely to be correlated. Intra prediction can be performed in spatial or transform domain, i.e., either sample values or transform coefficients can be predicted. Intra prediction is typically exploited in intra coding, where no inter prediction is applied.

5

One outcome of the coding procedure is a set of coding parameters, such as motion vectors and quantized transform coefficients. Many parameters can be entropy-coded more efficiently if they are predicted first from spatially or temporally neighboring parameters. For example, a motion vector may be predicted from spatially adjacent motion vectors and only the difference relative to the motion vector predictor may be coded. Prediction of coding parameters and intra prediction may be collectively referred to as in-picture prediction.

10

15

The decoder reconstructs the output video by applying prediction means similar to the encoder to form a predicted representation of the pixel blocks (using the motion or spatial information created by the encoder and stored in the compressed representation) and prediction error decoding (inverse operation of the prediction error coding recovering the quantized prediction error signal in spatial pixel domain). After applying prediction and prediction error decoding means, the decoder sums up the prediction and prediction error signals (pixel values) to form the output video frame. The decoder (and encoder) can also apply additional filtering means to improve the quality of the output video before passing it for display and/or storing it as prediction reference for the forthcoming frames in the video sequence.

20

25

In video codecs, the motion information may be indicated with motion vectors associated with each motion compensated image block. Each of these motion vectors represents the displacement of the image block in the picture to be coded (in the encoder side) or decoded (in the decoder side) and the prediction source block in one of the previously coded or decoded pictures. In order to represent motion vectors efficiently, those may be coded differentially with respect to block specific predicted motion vectors. In video codecs, the predicted motion vectors may be created in a predefined way, for example calculating the median of the encoded or decoded motion vectors of the adjacent blocks. Another way to create motion vector predictions is to generate a list of candidate predictions from adjacent blocks and/or co-located blocks in temporal reference pictures and signaling the chosen candidate as the motion vector predictor. In addition to predicting the motion vector values, the reference index of previously coded/decoded picture can be predicted.

30

35

The reference index is typically predicted from adjacent blocks and/or or co-located blocks in temporal reference picture. Moreover, high efficiency video codecs can employ an additional motion information coding/decoding mechanism, often called merging/merge mode, where all the motion field information, which includes motion vector and corresponding reference picture index for each available reference picture list, is predicted and used without any modification/correction. Similarly, predicting the motion field information may be carried out using the motion field information of adjacent blocks and/or co-located blocks in temporal reference pictures and the used motion field information is signaled among a list of motion field candidate list filled with motion field information of available adjacent/co-located blocks.

In video codecs the prediction residual after motion compensation may be first transformed with a transform kernel (like DCT) and then coded. The reason for this is that often there still exists some correlation among the residual and transform can in many cases help reduce this correlation and provide more efficient coding.

Video encoders may utilize Lagrangian cost functions to find optimal coding modes, e.g. the desired Macroblock mode and associated motion vectors. This kind of cost function uses a weighting factor λ to tie together the (exact or estimated) image distortion due to lossy coding methods and the (exact or estimated) amount of information that is required to represent the pixel values in an image area:

$$C = D + \lambda R$$

where C is the Lagrangian cost to be minimized, D is the image distortion (e.g. Mean Squared Error) with the mode and motion vectors considered, and R the number of bits needed to represent the required data to reconstruct the image block in the decoder (including the amount of data to represent the candidate motion vectors).

Video coding specifications may enable the use of supplemental enhancement information (SEI) messages or alike. Some video coding specifications include SEI network abstraction layer (NAL) units, and some video coding specifications contain both prefix SEI NAL units and suffix SEI NAL units, where the former type can start a picture unit or alike and the latter type can end a picture unit or alike. An SEI NAL unit contains one or more SEI messages, which are not required for the decoding of output pictures but may assist in related processes, such as picture output timing,

post-processing of decoded pictures, rendering, error detection, error concealment, and resource reservation. Several SEI messages are specified in H.264/AVC, H.265/HEVC, H.266/VVC, and H.274/VSEI standards, and the user data SEI messages enable organizations and companies to specify SEI messages for their own use. The standards may contain the syntax and semantics for the specified SEI messages but a process for handling the messages in the recipient might not be defined. Consequently, encoders may be required to follow the standard specifying a SEI message when they create SEI message(s), and decoders might not be required to process SEI messages for output order conformance. One of the reasons to include the syntax and semantics of SEI messages in standards is to allow different system specifications to interpret the supplemental information identically and hence interoperate. It is intended that system specifications can require the use of particular SEI messages both in the encoding end and in the decoding end, and additionally the process for handling particular SEI messages in the recipient can be specified.

The phrase along the bitstream (e.g. indicating along the bitstream) or along a coded unit of a bitstream (e.g. indicating along a coded tile) may be used in claims and described embodiments to refer to transmission, signaling, or storage in a manner that the "out-of-band" data is associated with but not included within the bitstream or the coded unit, respectively. The phrase decoding along the bitstream or along a coded unit of a bitstream or alike may refer to decoding the referred out-of-band data (which may be obtained from out-of-band transmission, signaling, or storage) that is associated with the bitstream or the coded unit, respectively. For example, the phrase along the bitstream may be used when the bitstream is contained in a container file, such as a file conforming to the ISO Base Media File Format, and certain file metadata is stored in the file in a manner that associates the metadata to the bitstream, such as boxes in the sample entry for a track containing the bitstream, a sample group for the track containing the bitstream, or a timed metadata track associated with the track containing the bitstream.

Image and video codecs may use a set of filters to enhance the visual quality of the predicted visual content and can be applied either in-loop or out-of-loop, or both. In the case of in-loop filters, the filter applied on one block in the currently-encoded frame will affect the encoding of another block in the same frame and/or in another frame which is predicted from the current frame. An in-loop filter can affect the bitrate and/or the visual quality. In fact, an enhanced block will cause a smaller residual

(difference between original block and predicted-and-filtered block), thus requiring less bits to be encoded. An out-of-the loop filter will be applied on a frame after it has been reconstructed, the filtered visual content won't be as a source for prediction, and thus it may only impact the visual quality of the frames that are output by the decoder.

Recently, neural networks (NNs) have been used in the context of image and video compression, by following mainly two approaches.

In one approach, NNs are used to replace one or more of the components of a traditional codec such as VVC/H.266. Here, term "traditional" refers to those codecs whose components and their parameters may not be learned from data. Examples of such components are:

- Additional in-loop filter, for example by having the NN as an additional in-loop filter with respect to the traditional loop filters.
- Single in-loop filter, for example by having the NN replacing all traditional in-loop filters.
- Intra-frame prediction.
- Inter-frame prediction.
- Transform and/or inverse transform.
- Probability model for the arithmetic codec.
- Etc.

Figure 1 illustrates examples of functioning of NNs as components of a traditional codec's pipeline, in accordance with an embodiment. In particular, Figure 1 illustrates an encoder, which also includes a decoding loop. Figure 1 is shown to include components described below:

- A luma intra pred block or circuit 101. This block or circuit performs intra prediction in the luma domain, for example, by using already reconstructed data from the same frame. The operation of the luma intra pred block or circuit 101 may be performed by a deep neural network such as a convolutional auto-encoder.

- A chroma intra pred block or circuit 102. This block or circuit performs intra prediction in the chroma domain, for example, by using already reconstructed data from the same frame. The chroma intra pred block or circuit 102 may perform cross-component prediction, for example, predicting chroma from luma. The operation of

the chroma intra pred block or circuit 102 may be performed by a deep neural network such as a convolutional auto-encoder.

- An intra pred block or circuit 103 and inter-pred block or circuit 104. These blocks or circuit perform intra prediction and inter-prediction, respectively. The intra pred block or circuit 103 and the inter-pred block or circuit 104 may perform the prediction on all components, for example, luma and chroma. The operations of the intra pred block or circuit 103 and inter-pred block or circuit 104 may be performed by two or more deep neural networks such as convolutional auto-encoders.

- A probability estimation block or circuit 105 for entropy coding. This block or circuit performs prediction of probability for the next symbol to encode or decode, which is then provided to the entropy coding module 112, such as the arithmetic coding module, to encode or decode the next symbol. The operation of the probability estimation block or circuit 105 may be performed by a neural network.

- A transform and quantization (T/Q) block or circuit 106. These are actually two blocks or circuits. The transform and quantization block or circuit 106 may perform a transform of input data to a different domain, for example, the FFT transform would transform the data to frequency domain. The transform and quantization block or circuit 106 may quantize its input values to a smaller set of possible values. In the decoding loop, there may be inverse quantization block or circuit and inverse transform block or circuit 113. One or both of the transform block or circuit and quantization block or circuit may be replaced by one or two or more neural networks. One or both of the inverse transform block or circuit and inverse quantization block or circuit 113 may be replaced by one or two or more neural networks.

- An in-loop filter block or circuit 107. Operations of the in-loop filter block or circuit 107 is performed in the decoding loop, and it performs filtering on the output of the inverse transform block or circuit, or anyway on the reconstructed data, in order to enhance the reconstructed data with respect to one or more predetermined quality metrics. This filter may affect both the quality of the decoded data and the bitrate of the bitstream output by the encoder. The operation of the in-loop filter block or circuit 107 may be performed by a neural network, such as a convolutional auto-encoder. In examples, the operation of the in-loop filter may be performed by multiple steps or filters, where the one or more steps may be performed by neural networks.

- A postprocessing filter block or circuit 108. The postprocessing filter block or circuit 108 may be performed only at decoder side, as it may not affect the encoding process. The postprocessing filter block or circuit 108 filters the reconstructed data output by the in-loop filter block or circuit 107, in order to enhance the reconstructed data. The postprocessing filter block or circuit 108 may be replaced by a neural network, such as a convolutional auto-encoder.

- A resolution adaptation block or circuit 109: this block or circuit may downsample the input video frames, prior to encoding. Then, in the decoding loop, the reconstructed data may be upsampled, by the upsampling block or circuit 110, to the original resolution. The operation of the resolution adaptation block or circuit 109 block or circuit may be performed by a neural network such as a convolutional auto-encoder.

- An encoder control block or circuit 111. This block or circuit performs optimization of encoder's parameters, such as what transform to use, what quantization parameters (QP) to use, what intra-prediction mode (out of N intra-prediction modes) to use, and the like. The operation of the encoder control block or circuit 111 may be performed by a neural network, such as a classifier convolutional network, or such as a regression convolutional network.

- An ME/MC block or circuit 114 performs motion estimation and/or motion compensation, which are two key operations to be performed when performing inter-frame prediction. ME/MC stands for motion estimation / motion compensation.

In another approach, commonly referred to as "end-to-end learned compression", NNs are used as the main components of the image/video codecs. In this second approach, there are two main options:

Option 1: re-use the video coding pipeline but replace most or all the components with NNs. Referring to Figure 2, it illustrates an example of modified video coding pipeline based on a neural network, in accordance with an embodiment. An example of neural network may include, but is not limited to, a compressed representation of a neural network. Figure 2 is shown to include following components:

- A neural transform block or circuit 202: this block or circuit transforms the output of a summation/subtraction operation 203 to a new representation of that data, which may have lower entropy and thus be more compressible.
- A quantization block or circuit 204: this block or circuit quantizes an input data 201 to a smaller set of possible values.
- An inverse transform and inverse quantization blocks or circuits 206. These blocks or circuits perform the inverse or approximately inverse operation of the transform and the quantization, respectively.
- An encoder parameter control block or circuit 208. This block or circuit may control and optimize some or all the parameters of the encoding process, such as parameters of one or more of the encoding blocks or circuits.
- An entropy coding block or circuit 210. This block or circuit may perform lossless coding, for example based on entropy. One popular entropy coding technique is arithmetic coding.
- A neural intra-codec block or circuit 212. This block or circuit may be an image compression and decompression block or circuit, which may be used to encode and decode an intra frame. An encoder 214 may be an encoder block or circuit, such as the neural encoder part of an auto-encoder neural network. A decoder 216 may be a decoder block or circuit, such as the neural decoder part of an auto-encoder neural network. An intra-coding block or circuit 218 may be a block or circuit performing some intermediate steps between encoder and decoder, such as quantization, entropy encoding, entropy decoding, and/or inverse quantization.
- A deep loop filter block or circuit 220. This block or circuit performs filtering of reconstructed data, in order to enhance it.
- A decode picture buffer block or circuit 222. This block or circuit is a memory buffer, keeping the decoded frame, for example, reconstructed frames 224 and enhanced reference frames 226 to be used for inter prediction.
- An inter-prediction block or circuit 228. This block or circuit performs inter-frame prediction, for example, predicts from frames, for example, frames 232, which are temporally nearby. An ME/MC 230 performs motion estimation and/or motion compensation, which are two key operations to be performed when performing inter-frame prediction. ME/MC stands for motion estimation / motion compensation.

Option 2: re-design the whole pipeline, as follows. An example of option 2 is described in detail in Figure 3:

- Encoder NN: performs a non-linear transform
- Quantization and lossless encoding of the encoder NN's output.
- Lossless decoding and dequantization.
- Decoder NN: performs a non-linear inverse transform.

5

Figure 3 depicts an encoder and a decoder NNs being parts of a neural auto-encoder architecture, in accordance with an example. In Figure 3, the Analysis Network 301 is an Encoder NN, and the Synthesis Network 302 is the Decoder NN, which may together be referred to as spatial correlation tools 303, or as neural auto-encoder.

10

In the Option 2, the input data 304 is analyzed by the Encoder NN (Analysis Network 301), which outputs a new representation of that input data. The new representation may be more compressible. This new representation may then be quantized, by a quantizer 305, to a discrete number of values. The quantized data is then lossless encoded, for example by an arithmetic encoder 306, thus obtaining a bitstream 307. The example shown in Figure 3 includes an arithmetic decoder 308 and an arithmetic encoder 306. The arithmetic encoder 306, or the arithmetic decoder 308, or the combination of the arithmetic encoder 306 and arithmetic decoder 308 may be referred to as arithmetic codec in some embodiments. On the decoding side, the bitstream is first lossless decoded, for example, by using the arithmetic codec decoder 308. The lossless decoded data is dequantized and then input to the Decoder NN, Synthesis Network 302. The output is the reconstructed or decoded data 309.

15

20

25

In case of lossy compression, the lossy steps may comprise the Encoder NN and/or the quantization.

In order to train this system, a training objective function (also called "training loss") may be utilized, which may comprise one or more terms, or loss terms, or simply losses. In one example, the training loss comprises a reconstruction loss term and a rate loss term. The reconstruction loss encourages the system to decode data that is similar to the input data, according to some similarity metric. Examples of reconstruction losses are:

30

35

- Mean squared error (MSE).
- Multi-scale structural similarity (MS-SSIM)

- Losses derived from the use of a pretrained neural network. For example, $\text{error}(f1, f2)$, where $f1$ and $f2$ are the features extracted by a pretrained neural network for the input data and the decoded data, respectively, and $\text{error}()$ is an error or distance function, such as L1 norm or L2 norm.
- 5 - Losses derived from the use of a neural network that is trained simultaneously with the end-to-end learned codec. For example, adversarial loss can be used, which is the loss provided by a discriminator neural network that is trained adversarially with respect to the codec, following the settings proposed in the context of Generative Adversarial Networks (GANs) and their
10 variants.

The rate loss encourages the system to compress the output of the encoding stage, such as the output of the arithmetic encoder. By “compressing”, we mean reducing the number of bits output by the encoding stage.

15 When an entropy-based lossless encoder is used, such as an arithmetic encoder, the rate loss typically encourages the output of the Encoder NN to have low entropy. Example of rate losses are the following:

- A differentiable estimate of the entropy.
- 20 - A sparsification loss, i.e., a loss that encourages the output of the Encoder NN or the output of the quantization to have many zeros. Examples are L0 norm, L1 norm, L1 norm divided by L2 norm.
- A cross-entropy loss applied to the output of a probability model, where the probability model may be a NN used to estimate the probability of the next
25 symbol to be encoded by an arithmetic encoder.

One or more of reconstruction losses may be used, and one or more of the rate losses may be used, as a weighted sum. The different loss terms may be weighted using different weights, and these weights determine how the final system performs
30 in terms of rate-distortion loss. For example, if more weight is given to the reconstruction losses with respect to the rate losses, the system may learn to compress less but to reconstruct with higher accuracy (as measured by a metric that correlates with the reconstruction losses). These weights may be considered to be hyper-parameters of the training session, and may be set manually by the person
35 designing the training session, or automatically for example by grid search or by using additional neural networks.

As shown in Figure 4, a neural network-based end-to-end learned video coding system may contain an encoder 401, a quantizer 402, a probability model 403, an entropy codec 420 (for example arithmetic encoder 405 / arithmetic decoder 406), a dequantizer 407, and a decoder 408. The encoder 401 and decoder 408 may be two neural networks, or mainly comprise neural network components. The probability model 403 may also comprise mainly neural network components. Quantizer 402, dequantizer 407 and entropy codec 420 may not be based on neural network components, but they may also comprise neural network components, potentially.

On encoder side, the encoder component 401 takes a video x 409 as input and converts the video from its original signal space into a latent representation that may comprise a more compressible representation of the input. In the case of an input image, the latent representation may be a 3-dimensional tensor, where two dimensions represent the vertical and horizontal spatial dimensions, and the third dimension represent the “channels” which contain information at that specific location. If the input image is a 128x128x3 RGB image (with horizontal size of 128 pixels, vertical size of 128 pixels, and 3 channels for the Red, Green, Blue color components), and if the encoder downsamples the input tensor by 2 and expands the channel dimension to 32 channels, then the latent representation is a tensor of dimensions (or “shape”) 64x64x32 (i.e., with horizontal size of 64 elements, vertical size of 64 elements, and 32 channels). Please note that the order of the different dimensions may differ depending on the convention which is used; in some cases, for the input image, the channel dimension may be the first dimension, so for the above example, the shape of the input tensor may be represented as 3x128x128, instead of 128x128x3. In the case of an input video (instead of just an input image), another dimension in the input tensor may be used to represent temporal information.

The quantizer component 402 quantizes the latent representation into discrete values given a predefined set of quantization levels. Probability model 403 and arithmetic codec component 420 work together to perform lossless compression for the quantized latent representation and generate bitstreams to be sent to the decoder side. Given a symbol to be encoded into the bitstream, the probability model 403 estimates the probability distribution of all possible values for that symbol based on a context that is constructed from available information at the current encoding/decoding state, such as the data that has already been encoded/decoded.

Then, the arithmetic encoder 405 encodes the input symbols to bitstream using the estimated probability distributions.

On the decoder side, opposite operations are performed. The arithmetic decoder 406 and the probability model 403 first decode symbols from the bitstream to recover the quantized latent representation. Then the dequantizer 407 reconstructs the latent representation in continuous values and pass it to decoder 408 to recover the input video/image. Note that the probability model 403 in this system is shared between the encoding and decoding systems. In practice, this means that a copy of the probability model 403 is used at encoder side, and another exact copy is used at decoder side.

In this system, the encoder 401, probability model 403, and decoder 408 may be based on deep neural networks. The system may be trained in an end-to-end manner by minimizing the following rate-distortion loss function:

$$L = D + \lambda R,$$

where D is the distortion loss term, R is the rate loss term, and λ is the weight that controls the balance between the two losses. The distortion loss term may be the mean square error (MSE), structure similarity (SSIM) or other metrics that evaluate the quality of the reconstructed video. Multiple distortion losses may be used and integrated into D , such as a weighted sum of MSE and SSIM. The rate loss term is normally the estimated entropy of the quantized latent representation, which indicates the number of bits necessary to represent the encoded symbols, for example, bits-per-pixel (bpp).

For lossless video/image compression, the system may contain only the probability model 403 and arithmetic encoder/decoder 405, 406. The system loss function contains only the rate loss, since the distortion loss is always zero (i.e., no loss of information).

Reducing the distortion in image and video compression is often intended to increase human perceptual quality, as humans are considered to be the end users, i.e. consuming/watching the decoded image. Recently, with the advent of machine learning, especially deep learning, there is a rising number of machines (i.e., autonomous agents) that analyze data independently from humans and that may

even take decisions based on the analysis results without human intervention. Examples of such analysis are object detection, scene classification, semantic segmentation, video event detection, anomaly detection, pedestrian tracking, etc. Example use cases and applications are self-driving cars, video surveillance cameras and public safety, smart sensor networks, smart TV and smart advertisement, person re-identification, smart traffic monitoring, drones, etc. When the decoded data is consumed by machines, a different quality metric shall be used instead of human perceptual quality. Also, dedicated algorithms for compressing and decompressing data for machine consumption are likely to be different than those for compressing and decompressing data for human consumption. The set of tools and concepts for compressing and decompressing data for machine consumption is referred to here as Video Coding for Machines.

It is likely that the receiver-side device has multiple “machines” or task neural networks (Task-NNs). These multiple machines may be used in a certain combination which is for example determined by an orchestrator sub-system. The multiple machines may be used for example in succession, based on the output of the previously used machine, and/or in parallel. For example, a video which was compressed and then decompressed may be analyzed by one machine (NN) for detecting pedestrians, by another machine (another NN) for detecting cars, and by another machine (another NN) for estimating the depth of all the pixels in the frames.

In this description, task machine and machine and task neural network are referred to interchangeably, and for such referral any process or algorithm (learned or not from data) which analyzes or processes data for a certain task is meant. In the rest of the description, other assumptions made regarding the machines considered in this disclosure may be specified in further details.

Also, it is to be noticed that terms “receiver-side” or “decoder-side” are used to refer to the physical or abstract entity or device which may contain one or more machines, and runs these one or more machines on encoded and eventually decoded video representation which may be encoded by another physical or abstract entity or device, the “encoder-side device”.

The encoded video data may be stored into a memory device, for example as a file. The stored file may later be provided to another device. Alternatively, the encoded video data may be streamed from one device to another.

Figure 5 is a general illustration of the pipeline of Video Coding for Machines. A VCM encoder 502 encodes the input video into a bitstream 504. A bitrate 506 may be computed 508 from the bitstream 504 in order to evaluate the size of the bitstream. A VCM decoder 510 decodes the bitstream output by the VCM encoder 502. In Figure 5, the output of the VCM decoder 510 is referred to as “Decoded data for machines” 512. This data may be considered as the decoded or reconstructed video. However, in some implementations of this pipeline, this data may not have same or similar characteristics as the original video which was input to the VCM encoder 502. For example, this data may not be easily understandable by a human by simply rendering the data onto a screen. The output of VCM decoder is then input to one or more task neural networks 514. In the figure, for the sake of illustrating that there may be any number of task-NNs 514, there are three example task-NNs, and a non-specified one (Task-NN X). The goal of VCM is to obtain a low bitrate while guaranteeing that the task-NNs still perform well in terms of the evaluation metric 516 associated to each task.

One of the possible approaches to realize video coding for machines is an end-to-end learned approach. In this approach, the VCM encoder and VCM decoder mainly consist of neural networks. Figure 6 illustrates an example of a pipeline for the end-to-end learned approach. The video is input to a neural network encoder 601. The output of the neural network encoder 601 is input to a lossless encoder 602, such as an arithmetic encoder, which outputs a bitstream 604. The lossless codec may be a probability model 603, both in the lossless encoder and in the lossless decoder, which predicts the probability of the next symbol to be encoded and decoded. The probability model 603 may also be learned, for example it may be a neural network. At decoder-side, the bitstream 604 is input to a lossless decoder 605, such as an arithmetic decoder, whose output is input to a neural network decoder 606. The output of the neural network decoder 606 is the decoded data for machines 607, that may be input to one or more task-NNs 608.

Figure 7 illustrates an example of how the end-to-end learned system may be trained. For the sake of simplicity, only one task-NN 707 is illustrated. A rate loss 705 may be computed from the output of the probability model 703. The rate loss 705 provides an approximation of the bitrate required to encode the input video data. A task loss 710 may be computed 709 from the output 708 of the task-NN 707.

The rate loss 705 and the task loss 710 may then be used to train 711 the neural networks used in the system, such as the neural network encoder 701, the probability model 703, the neural network decoder 706. Training may be performed by first computing gradients of each loss with respect to the neural networks that are contributing or affecting the computation of that loss. The gradients are then used by an optimization method, such as Adam, for updating the trainable parameters of the neural networks.

The machine tasks may be performed at decoder side (instead of at encoder side) for multiple reasons, for example because the encoder-side device does not have the capabilities (computational, power, memory) for running the neural networks that perform these tasks, or because some aspects or the performance of the task neural networks may have changed or improved by the time that the decoder-side device needs the tasks results (e.g., different or additional semantic classes, better neural network architecture). Also, there could be a customization need, where different clients would run different neural networks for performing these machine learning tasks.

Alternatively to an end-to-end trained codec, a video codec for machines can be realized by using a traditional codec such as H.266/VVC.

Alternatively, as described already above for the case of video coding for humans, another possible design may comprise using a traditional "base" codec, such as H.266/VCC, which additionally comprises one or more neural networks. In one possible implementation, the one or more neural networks may replace or be an alternative of one of the components of the traditional codec, such as:

- one or more in-loop filters;
- one or more intra-prediction modes;
- one or more inter-prediction modes;
- one or more transforms;
- one or more inverse transforms;
- one or more probability models, for lossless coding;
- one or more post-processing filters.

In another possible implementation, the one or more neural networks may function as an additional component, such as:

- one or more additional in-loop filters;

- one or more additional intra-prediction modes;
- one or more additional inter-prediction modes;
- one or more additional transforms;
- one or more additional inverse transforms;
- 5 - one or more additional probability models, for lossless coding;
- one or more additional post-processing filters.

Alternatively, another possible design may comprise using any codec architecture (such as a traditional codec, or a traditional codec which includes one or more neural
10 networks, or an end-to-end learned codec), and having a post-processing neural network, which adapts the output of the decoder so that it can be analyzed more effectively by one or more machines or task neural networks. For example, the encoder and decoder may be conformant to the H.266/VVC standard, a post-processing neural network takes the output of the decoder, and the output of the
15 post-processing neural network is then input to an object detection neural network. In this example, the object detection neural network is the machine or a task neural network.

Figure 8 illustrates an example including an encoder 810, a decoder 820, a post-processing filter 830, and a set of task-NNs 840. The encoder 810 and decoder 820
20 may represent a traditional image or video codec, such as a codec conformant with the VVC/H.266 standard, or may represent an end-to-end (E2E) learned image or video codec. The post-processing filter 830 may be a neural network based filter. The task-NNs 840 may be neural networks that perform different tasks with respect
25 to each other, wherein the tasks may comprise, for example, object detection, object segmentation, object tracking, etc.

In a VCM system, to serve target machine tasks better, the input media data may be pre-processed by a pre-processing filter before being compressed by a video
30 codec. There are different types of pre-processing filters. For example, pre-processing may include, but are not limited to, one or more of the following:

- a pre-processing filter may smooth the areas that are unimportant for the machine tasks. The smoothing may be performed according to an importance map given to the system;
- 35 • a pre-processing filter may enhance edges (e.g. increase the contrast over detected object edges);

- a pre-processing filter may enhance the video quality, such as increasing the contrast, denoising, removing spatial blur, removing motion blur, etc., for better machine performance;
- a pre-processing filter may enhance salient object in the media data;
- 5 • a pre-processing filter may obscure or conceal certain parts of the video for privacy or security reason.

10 A pre-processing filter may be content adaptive, for example, behaving differently according to the content to be processed. A system may contain multiple pre-processing filters and the filters are selected according to the content of the video.

15 In the current codecs, in particular, video codec for human consumption, the decoder has no information about the pre-processing filter being used. This system design greatly limits the performance of the codec when used for machine tasks.

20 The present embodiments comprise a VCM system with one or more pre-processing filters and one or more post-processing filters for one or more target machine tasks. The one or more pre-processing filters and the one or more post-processing filters may be jointly trained to optimize the performance of the one or more machine tasks together with the VCM encoder and decoder in the VCM system. When the one or more pre-processing filters and the one or more post-processing filters are NN-based filters and the VCM encoder and decoder is not differentiable, a NN-based end-to-end codec may substitute the VCM encoder and decoder during the training.

25 In this specification, term “VCM codec” refers to “VCM encoder and decoder”. Also, term “data codec” refers to the “data encoder and data decoder”.

30 Within some of the embodiments, the one or more pre-processing filters and the one or more post-processing filters may be configured at system deployment stage, session initialization stage or during the data the data encoding/decoding stage. The configuration of the filters may be based on the one or more machine tasks that the system intends to perform.

35 Within some of the embodiments, some information of the one or more pre-processing filters is transferred to the decoder side, and the decoder determines the one or more post-processing filters according to the received information about the one or more pre-processing filters.

Within some of the embodiments, a selector component may be applied to select the optimal one or more pre-processing filters and the one or more post-processing filters based on the content of that data and the target machine tasks.

5

Within some of the embodiments, the post-processing filters may be adapted to the content of the data and the selected one or more pre-processing filters.

Multiple pre-processing filters and post-processing filters

10

Figure 9 shows a VCM system with pre-processing filter and post-processing filter according to an example. A VCM encoder 910 comprises a pre-processing filter and a data encoder; and a VCM decoder 920 comprises a data decoder and a post-processing filter.

15

A VCM system may use more than one pre-processing filters and one or more post-processing filters to achieve better performance. The one or more pre-processing filters and the one or more post-processing filters may be traditional filters, for example, linear or non-linear transformations, or NN-based filters that are pre-trained using training data to optimize the system performance for target machine tasks.

20

A sequence of the one or more pre-processing filters or a sequence of the one or more post-processing filters may be applied. A many-to-many, one-to-many, or many-to-one configuration is possible. The outputs of the one or more post-processing filters may be combined by a linear function. A VCM encoder may include coefficients of the linear function in or along a bitstream. A VCM decoder may decode coefficients of the linear function from or along a bitstream. For example, the coefficients of the linear function may be signaled from the encoder to the decoder through a communication protocol, such as the Session Description Protocol (SDP) specified by the Internet Engineering Task Force (IETF). In another example, the coefficients may be encoded as or decoded from one or more SEI messages.

25

30

35

According to an embodiment, the one or more pre-processing filters and the one or more post-processing filters may be configured at system deployment stage, session initialization stage or during the data encoding/decoding stage. The

configuration of the filters may be based on the one or more target machine tasks. For example, in a person identification system, the one or more pre-processing filters and the one or more post-processing filters that enhance human objects and obscure other objects may be configured to be used in the system.

5

The one or more pre-processing filters and the one or more post-processing filters may not be pre-configured for the target machine tasks. According to another embodiment, the one or more post-processing filters may be selected by the VCM decoder based on the one or more pre-processing filters being selected and the target machine tasks. At the encoder side, one or more pre-processing filters are selected for the input data.

10

According to an embodiment, the VCM encoder may determine the one or more post-processing filters to be used and signals an identifier (ID) of the one or more post-processing filters to the VCM decoder. The signalling may be sequence-based, frame-based, or region-based when applied to video compression. The VCM encoder may signal an identifier of the one or more post-processing filters in or along the bitstream. The identifier may for example be encoded within or decoded from one or more SEI messages. The determination of the one or more post-processing filters may use a proxy task network. The system may perform the determination as follows:

15

20

- extracting features by the VCM encoder using the proxy task network from the input uncompressed data;
- extracting features by the VCM encoder using the proxy task network from the reconstructed data that are encoded and decoded by the VCM codecs configured with the one or more selected pre-processing filters and different configurations of the one or more post-processing filters;
- selecting by the VCM encoder the configuration of the one or more post-processing filters, where the extracted features are the closest to the features of the input uncompressed data.

25

30

According to another embodiment, the VCM encoder may signal an identifier (ID) of the selected one or more pre-processing filters to the VCM decoder. The VCM encoder may signal an identifier of the one or more pre-processing filters in or along the bitstream. The identifier may for example be encoded within one or more SEI messages. In an embodiment, a VCM decoder decodes an identifier of the one or more pre-processing filters from or along the bitstream. The identifier may for

35

example be decoded from one or more SEI messages. The VCM decoder determines the one or more optimal post-processing filters according to the machine tasks and the one or more pre-processing filters that have been applied in the encoder side according to the decoded identifier. The determination may be based on the rules that are pre-configured in the system or using a learning-based approach.

An identifier of the one or more pre-processing filters and/or post-processing filters may be any identifier, such as any of the following:

- a Uniform Resource Identifier (URI),
- a universally unique identifier (UUID),
- a hash value (such as MD5 checksum) derived from a representation of the one or more post-processing filters, wherein the representation may for example comprise data bytes of the coefficients of the filter(s) or be the compressed representation of the filter(s) in a pre-defined or indicated compression format;
- a registered identification number, value, or character string (e.g. maintained by a registration authority).

According to another embodiment, the VCM encoder may transfer a new post-processing filter or the weight update to an existing post-processing filter to the VCM decoder. The VCM decoder may store the new post-processing filter or the weight update of an existing post-processing filter in storage and assign an ID to it so that it can be used later. The weight or the weight update of the post-processing filter, or URI of the weight or the weight update may be carried by SEI messages.

According to yet another embodiment, the VCM encoder may signal some information about the selected one or more pre-processing filters. This information may assist the VCM decoder in determining the one or more optimal post-processing filters. The signalling may be carried in or along the bitstream. The signalling may for example be encoded as or decoded from one or more SEI messages. The information may include, but is not necessarily limited to, one or more of the following:

- A purpose of the one or more pre-processing filters, for example, denoising, object enhancement, background removal, privacy preserving, etc.;
- A type of the one or more pre-processing filters, such as a Wiener filter or a neural network;

- A topology of the one or more pre-processing filters. The topology may be dependent on the type of the filter(s), and be defined e.g. as a number of taps for a linear filter or as an identifier of a neural network topology;
- Values for the parameters of the one or more pre-processing filters. The parameters may be dependent on the type and topology of the filter(s). The parameters may for example comprise the filter tap values of a linear filter, coefficient values for a Wiener filter (e.g. represented as or similarly to an adaptive loop filter (ALF) adaptation parameter set of VVC) or the weights of a neural network. The parameters may be indicated within the information or an identifier identifying pre-defined values of the parameters may be indicated within the information. For example, the information may comprise the weights of the selected one or more NN-based pre-processing filters of the URI(S) indicating the weights.

At the VCM decoder side, a selector module may be used to select the one or more post-processing filters based on the information received from the encoder. The selector component may be a NN-based component or a selection logic such as a decision tree. The selector component may be offline trained using a training data set. The training may use a proxy network or the task NN directly.

Content adaptive selection of pre-processing filters and post-processing filters

Figure 10 shows an example of a system with multiple pre-processing filters, multiple post-processing filters and selector modules 1010, 1040 at the VCM encoder and VCM decoder side. The pre-processing filters and the post-processing filters may be selected according to the content of the input data. For example, in a surveillance system, when the content contains human objects, the one or more pre-processing filters and the one or more post-processing filters that enhance human objects, may be selected to process the data. Content adaptive selector modules 1010, 1040 may be used at both the VCM encoder side and the VCM decoder side. The content adaptive pre-processing filter and post-processing filter selector may be NN-based modules and trained using a training dataset.

Adaptive post-processing filters

This chapter relates to methods for the one or more post-processing filters to be adapted to the one or more pre-processing filters used at the VCM encoder side.

It is assumed that the one or more pre-processing filters are available at the VCM decoder side. The one or more pre-processing filters may be included in the VCM decoder package or received from the encoder, for example, via one or more SEI messages.

Figure 11 illustrates the components, the data and the loss function at the VCM decoder side for the adaptation training. At the adaptation stage, the VCM encoder sends data x without applying any pre-processing filter to the VCM decoder. At the VCM decoder side, the VCM decoder first applies the one or more pre-processing filters 1110 to data y , which is the output of the data decoder in the VCM decoder and gets filtered data $y1$. Next, the VCM decoder applies the post-processing filter 1120 to data $y1$, and gets reconstructed data z . As a following step, the decoder optimizes the one or more post-processing filters by minimizing a loss function defined for data y and z . The loss function can be L1 or L2 loss or a loss function that involves a proxy network, for example, the L1 or L2 loss of the extracted features between data y and data z using the proxy network. The encoder may signal the hyper-parameters, for example, learning rate and the number of iterations, for the adaptation training.

The updated one or more post-processing filters may be stored to a storage at the VCM decoder side, and may be used later without the adaptation procedure.

After the adaptation is completed, the VCM decoder may apply the updated one or more post-processing filters when the one or more pre-processing filters are applied to the input data at the VCM encoder side. The VCM encoder may initiate the fine-tuning again whenever necessary.

According to another embodiment, the VCM encoder may generate adaptation data which is a collage of multiple different contents relevant for the targeted machine tasks. The VCM decoder is configured to optimize its one or more post-processing filters based on the defined loss function when applied to the adaptation data. According to another embodiment, such a collage content may be done using sub-picture encoding features of the encoder and additional metadata may be signaled using SEI messages or out of band. A sub-picture may be an isolated region, which

may be defined as a picture region that is allowed to depend only on the corresponding isolated region in reference pictures and does not depend on any other picture regions in the current picture or in the reference pictures. The corresponding isolated region in reference pictures may be for example the picture region that collocates with the isolated region in a current picture. A coded isolated region may be decoded without the presence of any picture regions of the same coded picture.

According to another embodiment, at the adaptation stage, the VCM encoder may send some measurements of the input data and compress the data without applying any pre-processing filters. These measurements may include:

- values of a set of selected pixels from the input data. The pixel may be selected according to a pre-determined pattern or randomly selected. When the pixels are randomly selected, the random seed may be pre-determined by the system or signaled from the VCM encoder to the VCM decoder;
- one or more selected patches from the input data;
- one or more random projections from the input data.

The VCM decoder may optimize the parameters of the one or more post-processing filters by minimizing the difference between the same measurements of the filtered output data and the received measurements of the input data. The VCM encoder may signal the hyper-parameters such as the learning rate and the number of iterations for the adaptation training.

Figure 12 illustrates the components, the data, the calculation of the measurements according to measurement functions 1220, 1230, and the loss function used for the adaptation training in this embodiment. In Figure 12, x is the input data to the VCM system, y is the input data to the one or more post-processing filters 1210, z is the output of the one or more post-processing filters 1210, $m(x)$ is the measurement of data x determined by the measurement function 1220, $m(z)$ is the measurement of data z determined by the measurement function 1230, and $Loss(m(x), m(z))$ is the loss function for measurements $m(x)$ and $m(z)$.

Figure 13 shows yet another embodiment, where the VCM decoder first applies one or more post-processing filters 1310 to data y to get the filtered output data z . Next, the VCM decoder may apply the one or more pre-processing filters 1320 to data z , and get data y_2 . As a following step, the parameters of the one or more post-

processing filters 1310 may be optimized to minimize a loss function defined for data y and y_2 . The VCM encoder may signal the hyper-parameters, such as the learning rate and the number of iterations for the adaptation training.

- 5 This method may be combined with the previous method where the one or more measurements are sent from the VCM encoder to the VCM decoder. The adaptation training may be performed by optimizing the weighted sum of $loss(y, y_2)$ and $loss(m(x), m(z))$.
- 10 The method for encoding according to an embodiment is shown in Figure 14. The method generally comprises receiving 1410 input media data; pre-processing 1420 the input media data by one or more pre-processing filters; compressing 1430 the pre-processed input media data by a video codec; and including 1440 information characterizing the one or more pre-processing filters in or along the compressed media data. Each of the steps can be implemented by a respective module of a computer system.

An apparatus according to an embodiment comprises means for receiving input media data; means for pre-processing the input media data by one or more pre-processing filters; means for compressing the pre-processed input media data by a video codec; and means for including information characterizing the one or more pre-processing filters in or along the compressed media data. The means comprises at least one processor, and a memory including a computer program code, wherein the processor may further comprise processor circuitry. The memory and the computer program code are configured to, with the at least one processor, cause the apparatus to perform the method of Figure 14 according to various embodiments.

- 30 The method for encoding according to an embodiment is shown in Figure 15. The method generally comprises receiving 1510 media data; decompressing 1520 the compressed media data; decoding 1530 information characterizing one or more pre-processing filters from or along the compressed media data, the one or more pre-processing filters having been applied to input media data in generation of the compressed media data; selecting 1540 one or more post-processing filters on the basis of the information characterizing the one or more pre-processing filters; post-processing 1550 the decompressed input media data by one or more post-

processing filters; and transferring 1560 the post-processed decompressed data for analysis at one or more task neural networks.

5 An apparatus according to an embodiment comprises means for receiving media data; means for decompressing the compressed media data; means for decoding information characterizing one or more pre-processing filters from or along the compressed media data, the one or more pre-processing filters having been applied to input media data in generation of the compressed media data; means for selecting one or more post-processing filters on the basis of the information
10 characterizing the one or more pre-processing filters; means for post-processing the decompressed input media data by one or more post-processing filters; and means for transferring the post-processed decompressed data for analysis at one or more task neural networks. The means comprises at least one processor, and a memory including a computer program code, wherein the processor may further comprise
15 processor circuitry. The memory and the computer program code are configured to, with the at least one processor, cause the apparatus to perform the method of Figure 15 according to various embodiments.

Figure 16 illustrates an example of an apparatus. The apparatus is a user equipment
20 for the purposes of the present embodiments. The apparatus 90 comprises a main processing unit 91, a memory 92, a user interface 94, a communication interface 93. The apparatus according to an embodiment, shown in Figure 15, may also comprise a camera module 95. Alternatively, the apparatus may be configured to receive image and/or video data from an external camera device over
25 a communication network. The memory 92 stores data including computer program code in the apparatus 90. The computer program code is configured to implement the method according various embodiments by means of various computer modules. The camera module 95 or the communication interface 93 receives data, in the form of images or video stream, to be processed by the
30 processor 91. The communication interface 93 forwards processed data, i.e. the image file, for example to a display of another device, such a virtual reality headset. When the apparatus 90 is a video source comprising the camera module 95, user inputs may be received from the user interface.

35 The various embodiments can be implemented with the help of computer program code that resides in a memory and causes the relevant apparatuses to carry out the method. For example, a device may comprise circuitry and electronics for handling,

receiving and transmitting data, computer program code in a memory, and a processor that, when running the computer program code, causes the device to carry out the features of an embodiment. Yet further, a network device like a server may comprise circuitry and electronics for handling, receiving and transmitting data, computer program code in a memory, and a processor that, when running the computer program code, causes the network device to carry out the features of various embodiments.

If desired, the different functions discussed herein may be performed in a different order and/or concurrently with other. Furthermore, if desired, one or more of the above-described functions and embodiments may be optional or may be combined.

Although various aspects of the embodiments are set out in the independent claims, other aspects comprise other combinations of features from the described embodiments and/or the dependent claims with the features of the independent claims, and not solely the combinations explicitly set out in the claims.

It is also noted herein that while the above describes example embodiments, these descriptions should not be viewed in a limiting sense. Rather, there are several variations and modifications, which may be made without departing from the scope of the present disclosure as, defined in the appended claims.

Claims:

1. An apparatus for encoding comprising:
 - means for receiving input media data;
 - 5 - means for pre-processing the input media data by one or more pre-processing filters;
 - means for compressing the pre-processed input media data by a video codec; and
 - 10 - means for including information characterizing the one or more pre-processing filters in or along the compressed media data.
2. The apparatus according to claim 1, wherein the pre-processing filter is configured to perform one or more of the following: smoothing areas that are unimportant for machine tasks; enhancing video quality for better machine performance; enhancing salient objects in the media data; obscuring or concealing certain parts of a video.
3. The apparatus according to claim 1 or 2, wherein the pre-processing filter is content adaptive.
4. The apparatus according to claim 1 or 2 or 3, wherein the information characterizing the one or more pre-processing filters comprises one or more of the following:
 - a purpose of said one or more pre-processing filters;
 - 25 - a type of said one or more pre-processing filters;
 - a topology of the one or more pre-processing filters;
 - values for parameters of said one or more pre-processing filters.
5. The apparatus according to any of the claims 1 to 4, further comprising one or more selector modules configured to select the optimal one or more pre-processing filters based on the content of the input media data and/or target machine tasks and/or the information.
6. The apparatus according to any of the claims 1 to 5, wherein said one or more pre-processing filters are any combination of the following: linear or non-linear transformation filters; neural network based filters.

7. The apparatus according to any of the claims 1 to 6, further comprising means for generating adaptation data to be used for training post-processing filters at a decoder.

5 8. An apparatus for decoding, comprising

- means for receiving compressed media data;
- means for decompressing the compressed media data;
- means for decoding information characterizing one or more pre-processing filters from or along the compressed media data, the one or more pre-processing filters having been applied to input media data in generation of the compressed media data;
- means for selecting one or more post-processing filters on the basis of the information characterizing the one or more pre-processing filters;
- means for post-processing the decompressed input media data by one or more post-processing filters; and
- means for transferring the post-processed decompressed data for analysis at one or more task neural networks.

20 9. The apparatus according to claim 8, further comprising means for adapting said one or more post-processing filters by minimizing a loss function.

25 10. The apparatus according to claim 9, wherein the loss function has been defined for a data input to one or more pre-processing filters and reconstructed data from one or more post-processing filters

11. The apparatus according to claim 9, wherein the loss function has been defined for a measurement of an input data and a measurement of an output from one or more post-processing filters.

30 12. The apparatus according to claim 9, wherein the loss function has been defined for a data input to one or more post-processing filters and data output from one or more pre-processing filters.

35 13. The apparatus according to any of the claims 8 to 12, wherein said one or more post-processing filters are any combination of the following: linear or non-linear transformation filters; neural network based filters.

14. A method for encoding, comprising:

- receiving input media data;
- pre-processing the input media data by one or more pre-processing filters;
- compressing the pre-processed input media data by a video codec; and
- including information characterizing the one or more pre-processing filters in or along the compressed media data.

15. A method for decoding, comprising

- receiving compressed media data;
- decompressing the compressed media data;
- decoding information characterizing one or more pre-processing filters from or along the compressed media data, the one or more pre-processing filters having been applied to input media data in generation of the compressed media data;
- selecting one or more post-processing filters on the basis of the information characterizing the one or more pre-processing filters;
- post-processing the decompressed input media data by one or more post-processing filters; and
- transferring the post-processed decompressed data for analysis at one or more task neural networks.

16. An apparatus comprising at least one processor, memory including computer program code, the memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the following:

- receive input media data;
- pre-process the input media data by one or more pre-processing filters;
- compress the pre-processed input media data by a video codec; and
- include information characterizing the one or more pre-processing filters in or along the compressed media data.

17. An apparatus comprising at least one processor, memory including computer program code, the memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the following:

- receive compressed media data;
- decompress the compressed media data;
- decode information characterizing one or more pre-processing filters from or along the compressed media data, the one or more pre-processing filters

having been applied to input media data in generation of the compressed media data;

- select one or more post-processing filters on the basis of the information characterizing the one or more pre-processing filters;
- 5 - post-process the decompressed input media data by one or more post-processing filters; and
- transfer the post-processed decompressed data for analysis at one or more task neural networks.

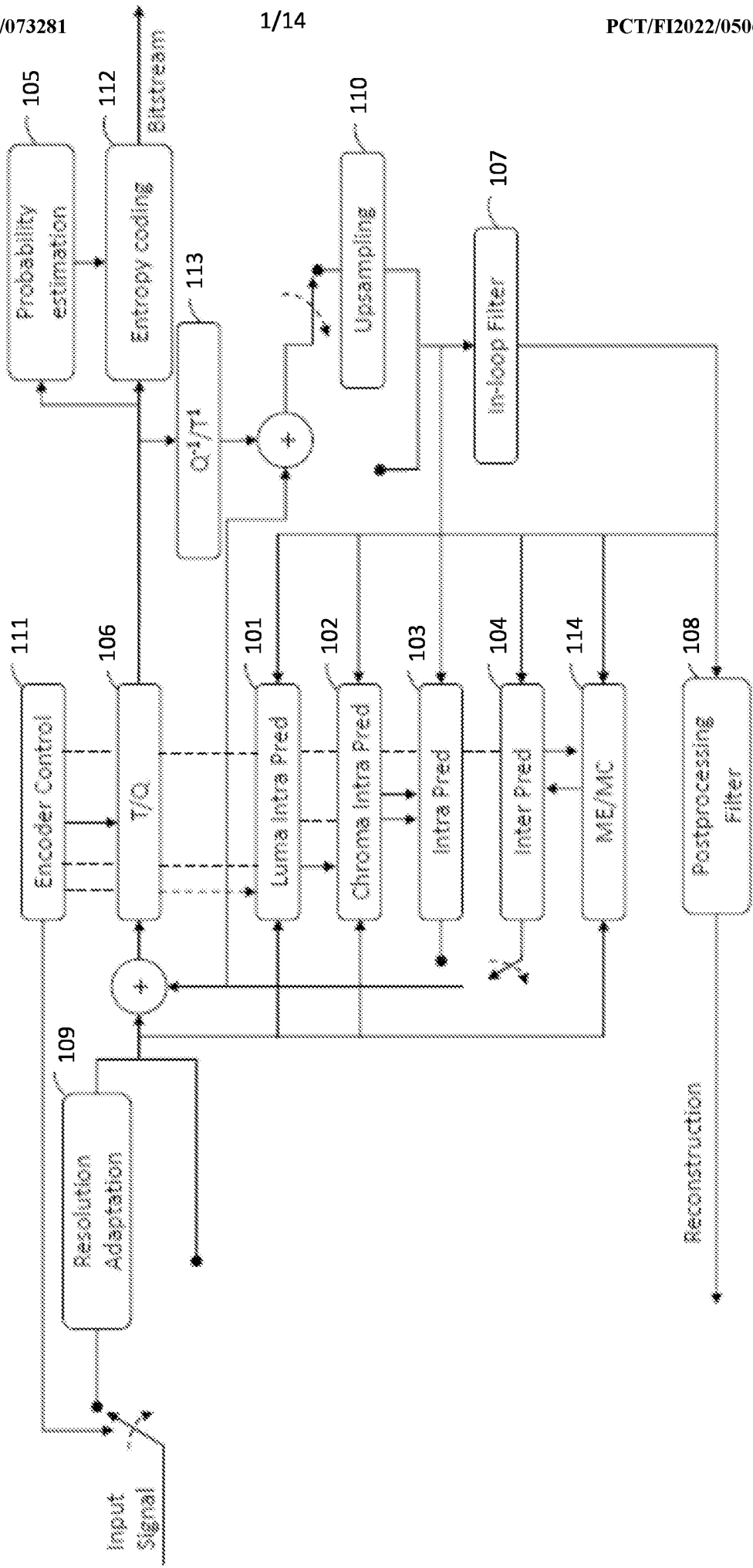


Fig. 1

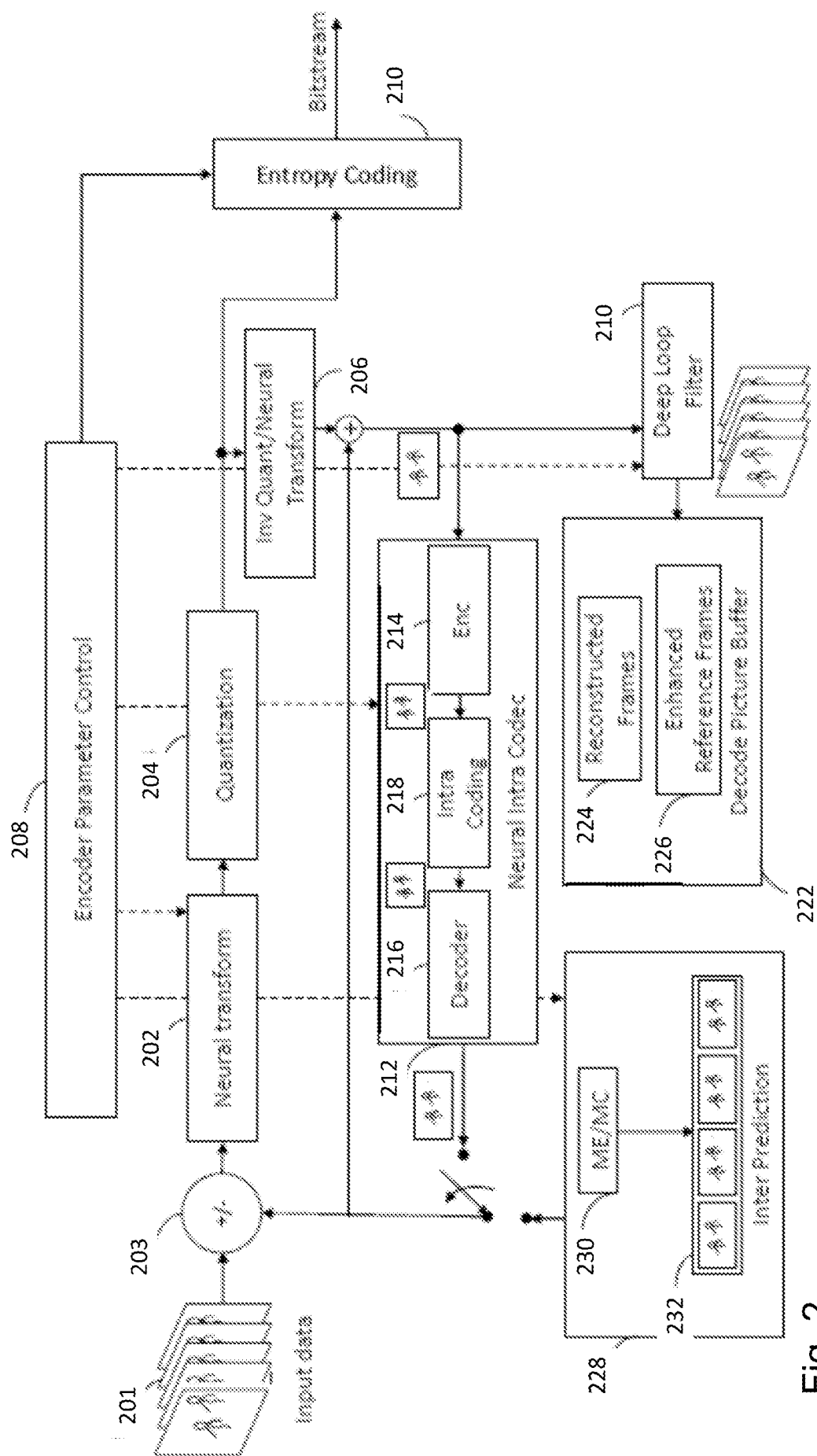


Fig. 2

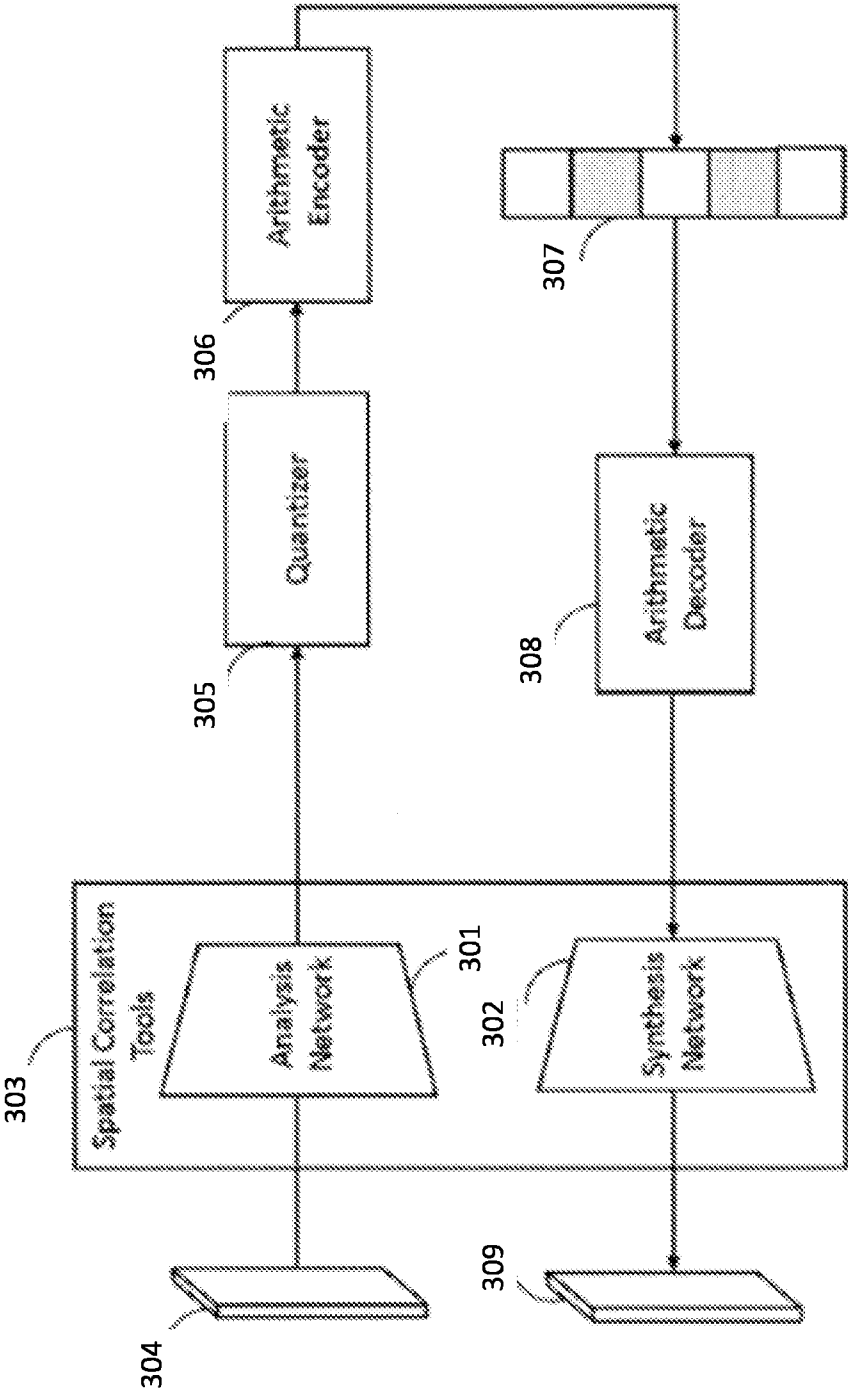


Fig. 3

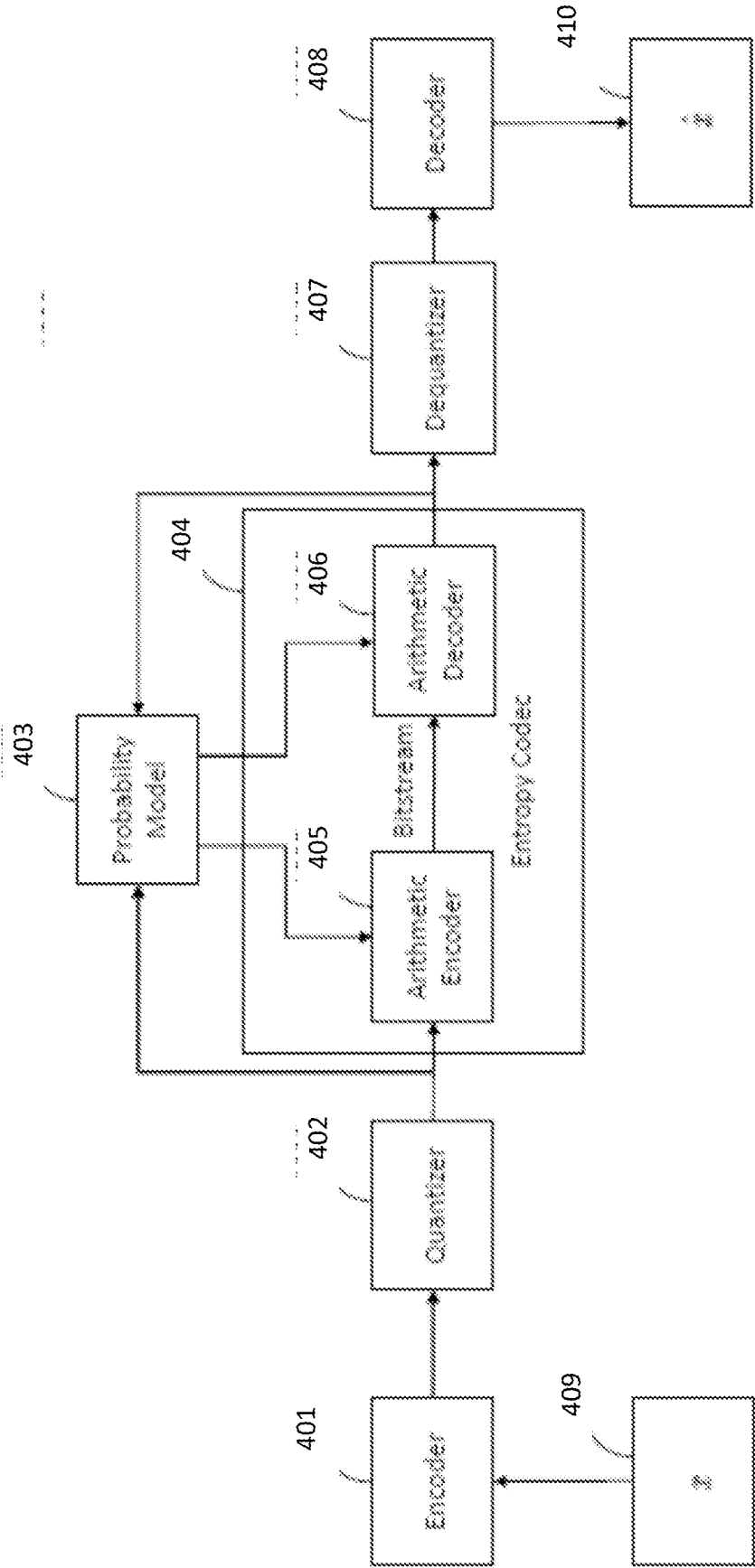


Fig. 4

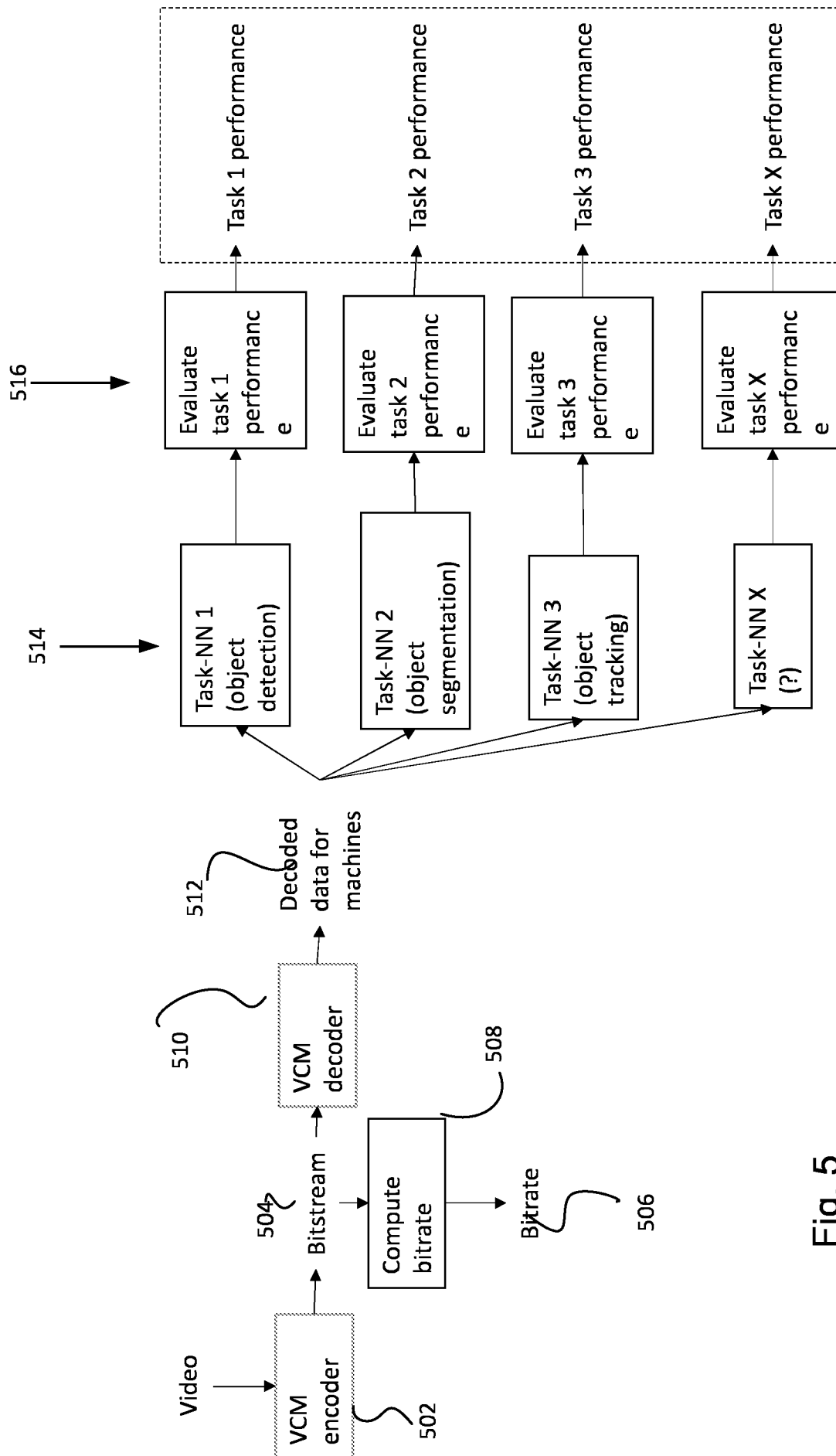


Fig. 5

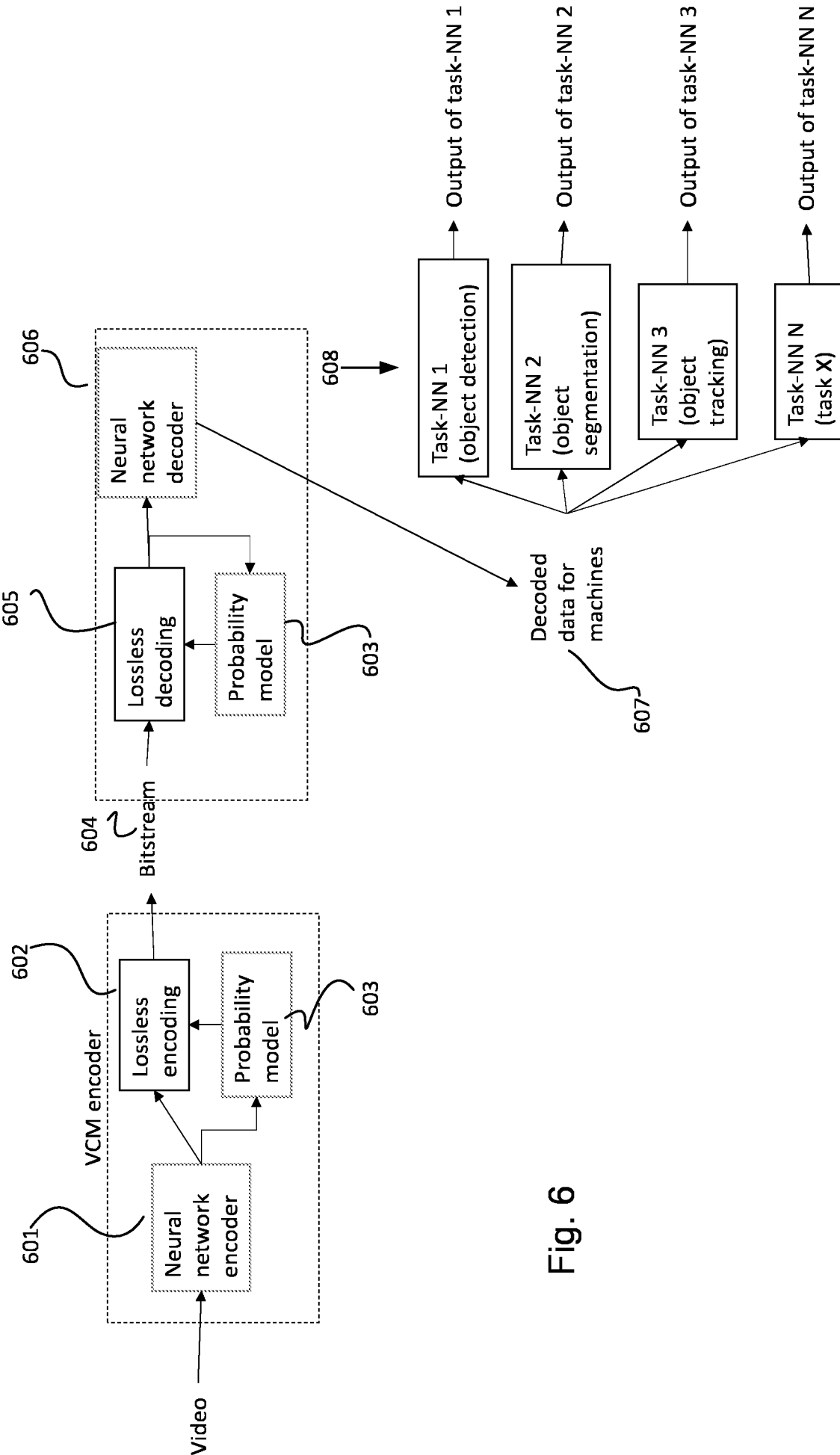


Fig. 6

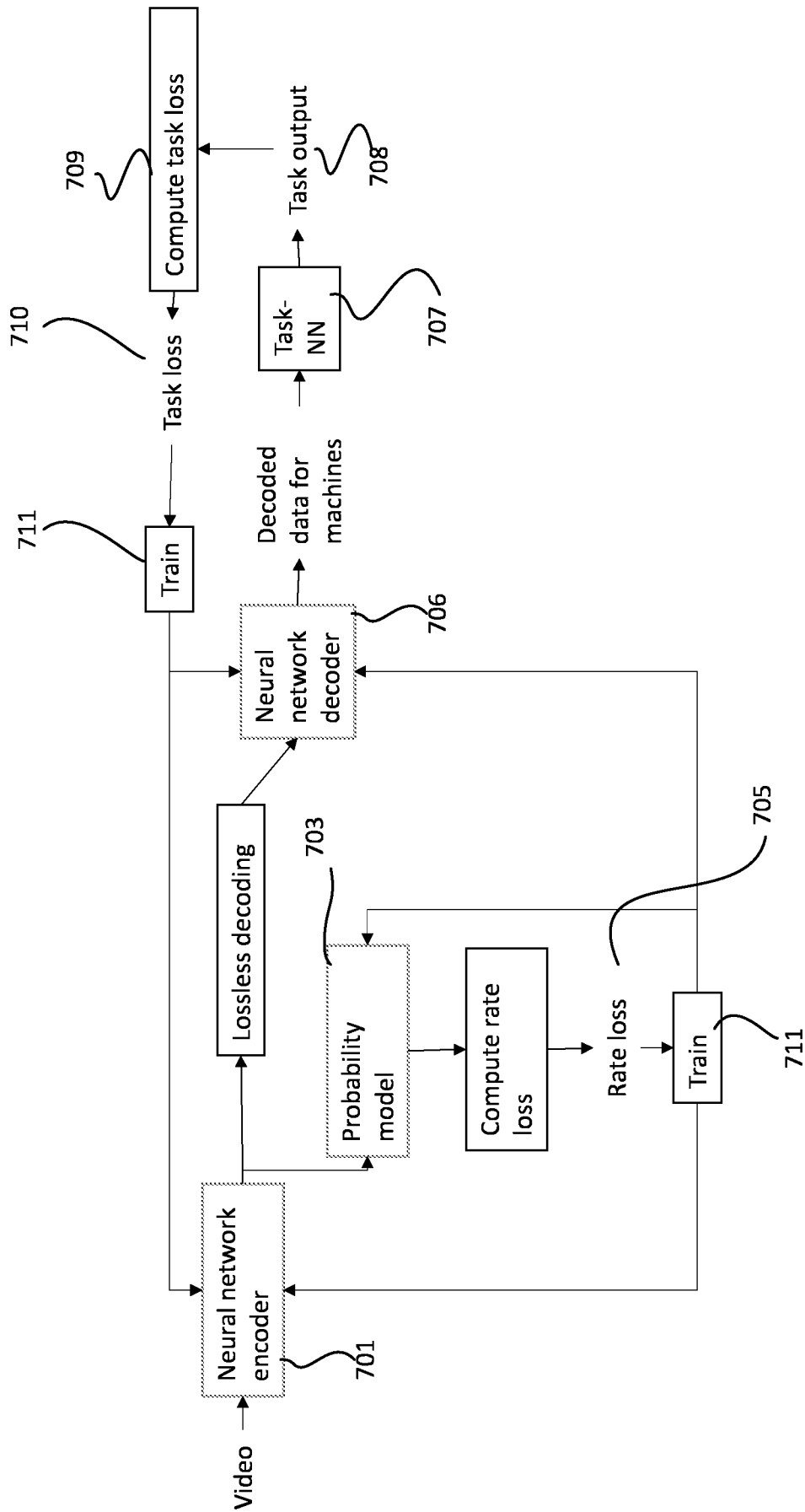


Fig. 7

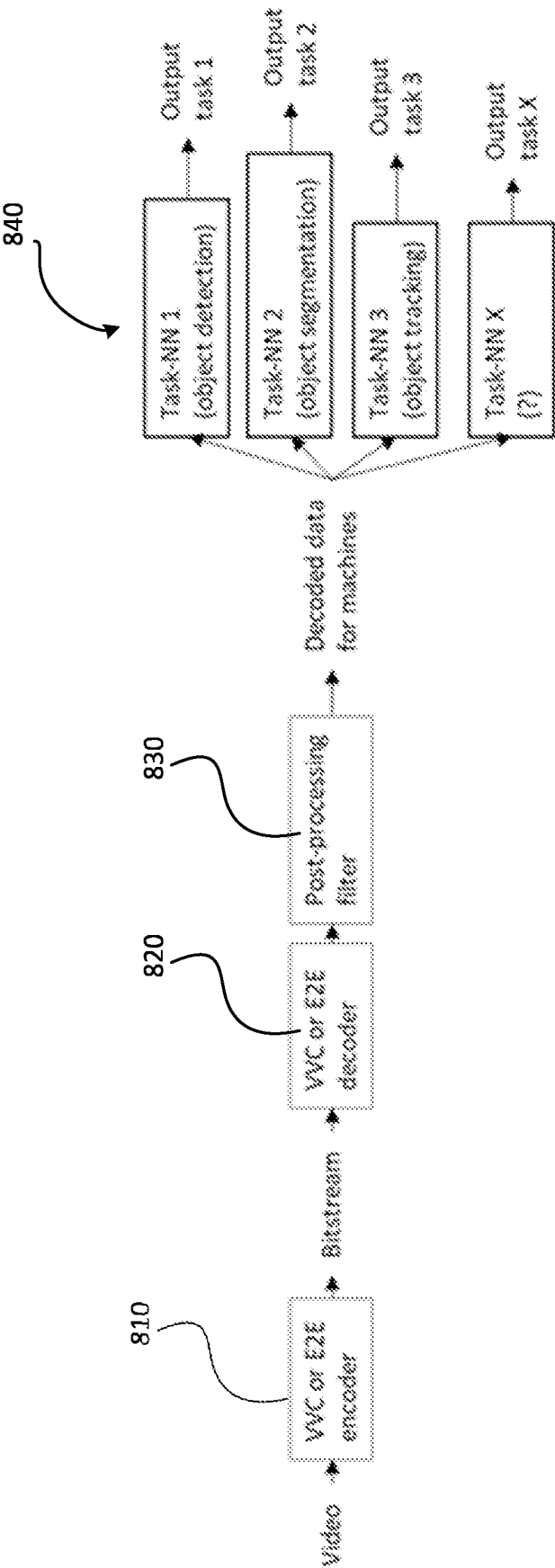


Fig. 8

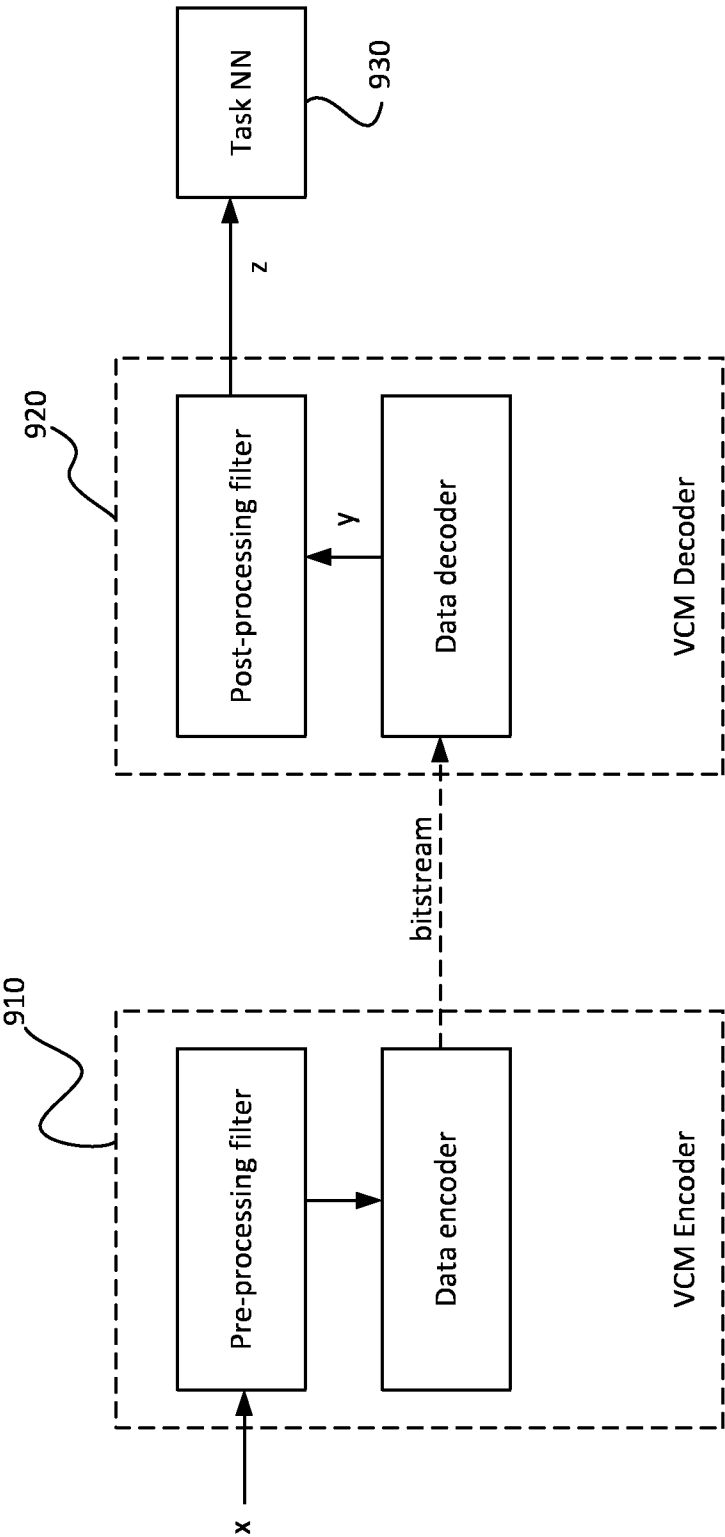


Fig. 9

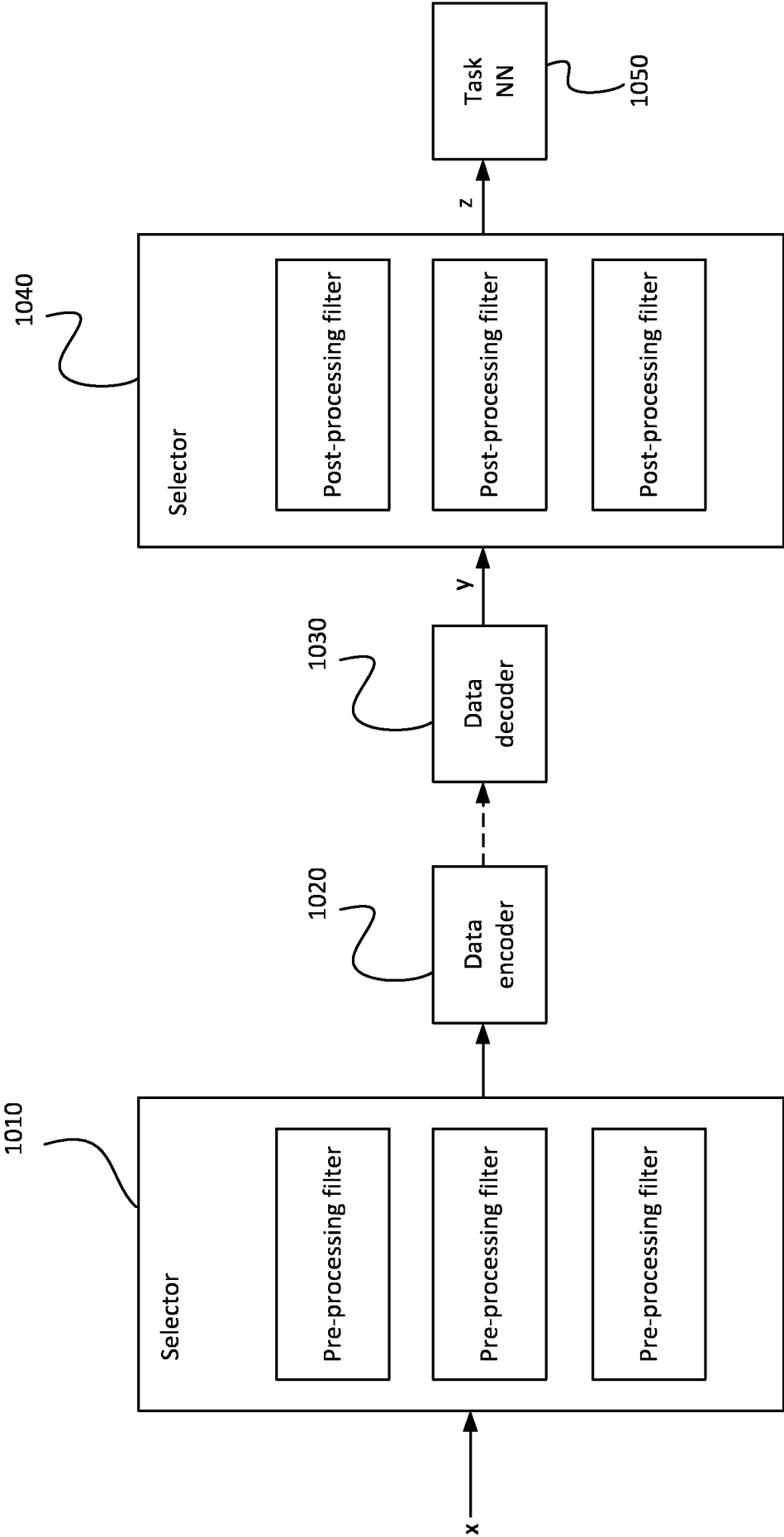


Fig. 10

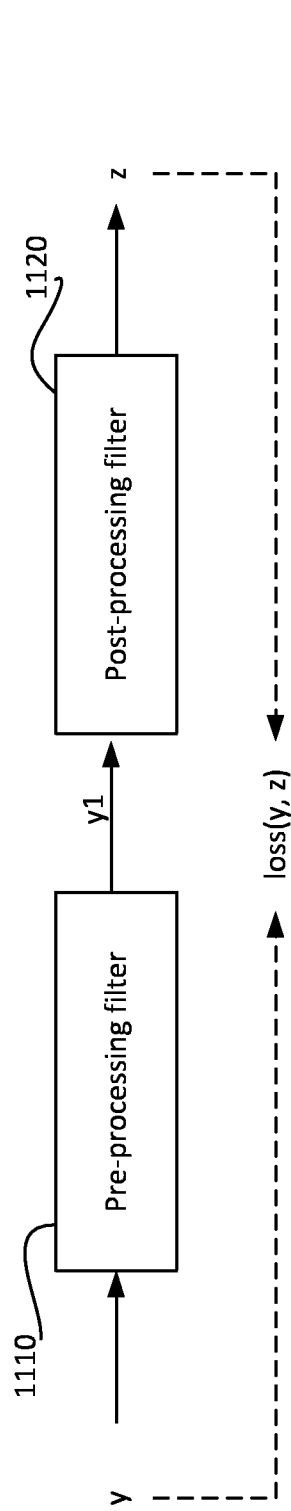


Fig. 11

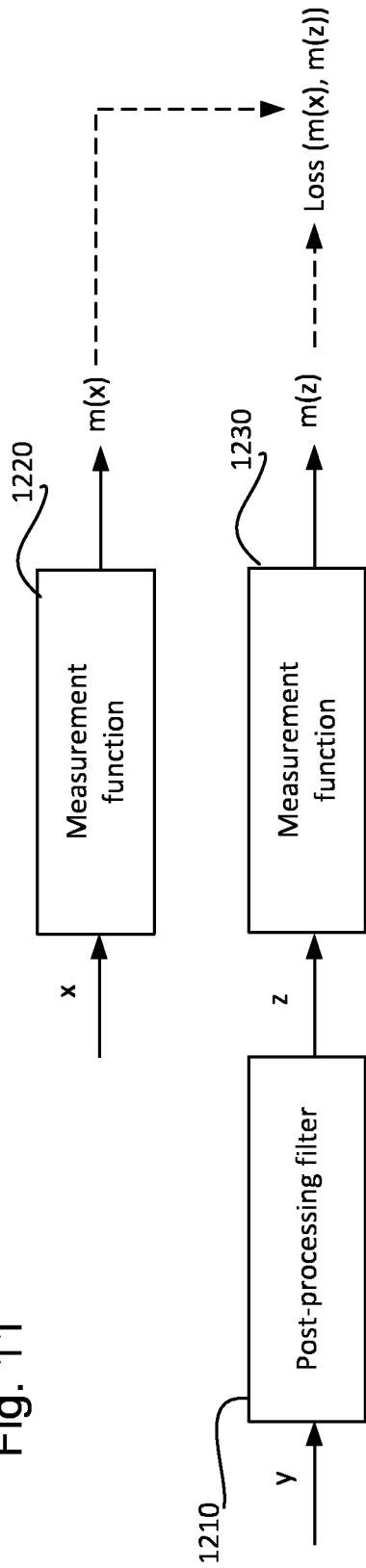


Fig. 12

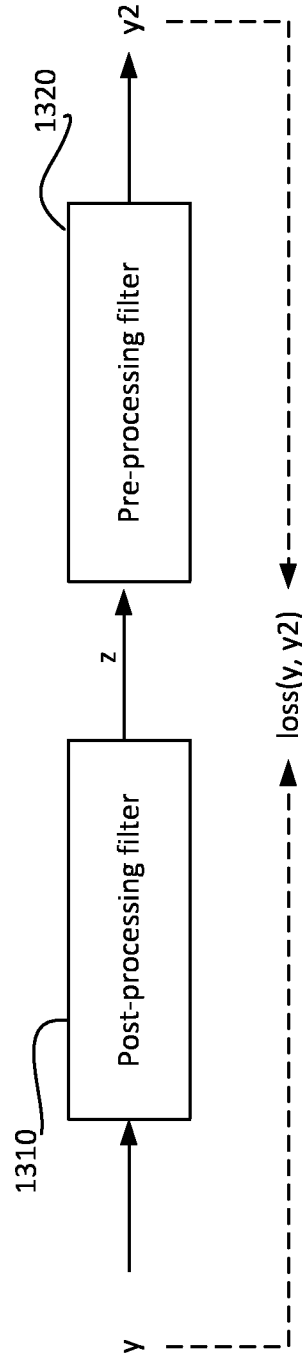


Fig. 13

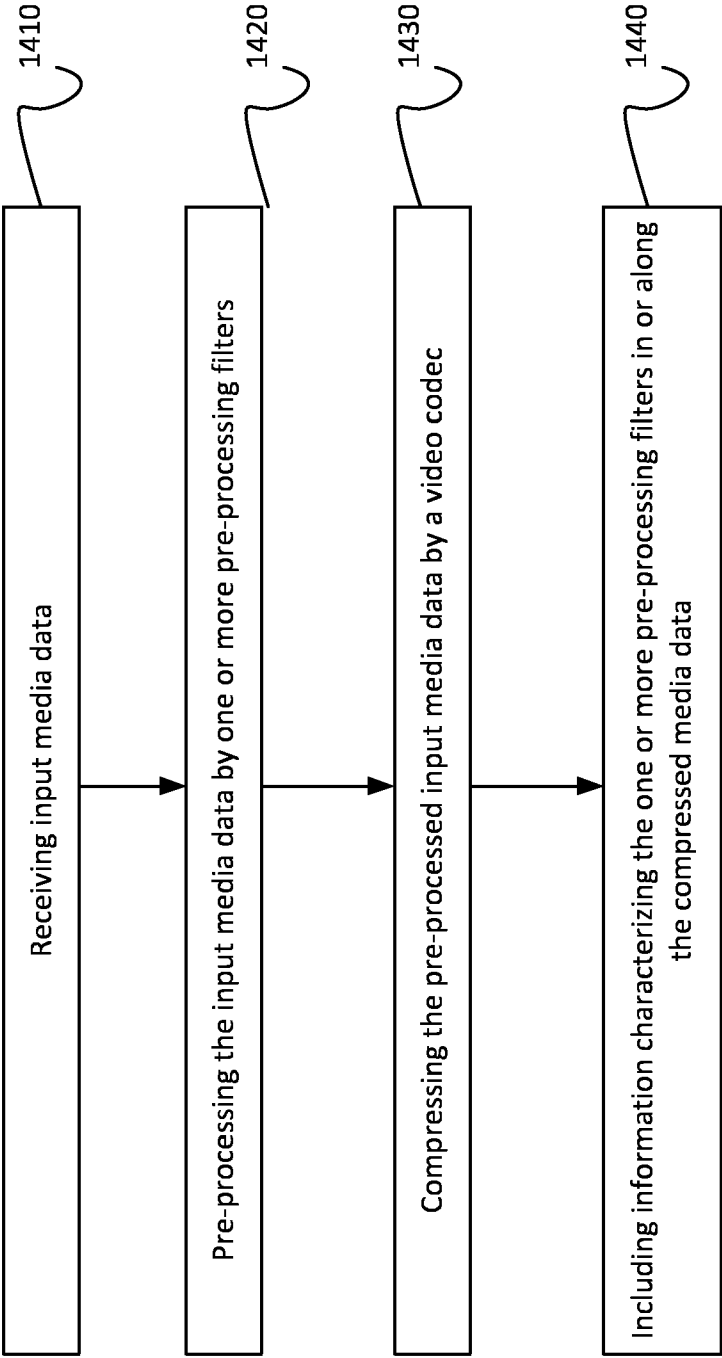


Fig. 14

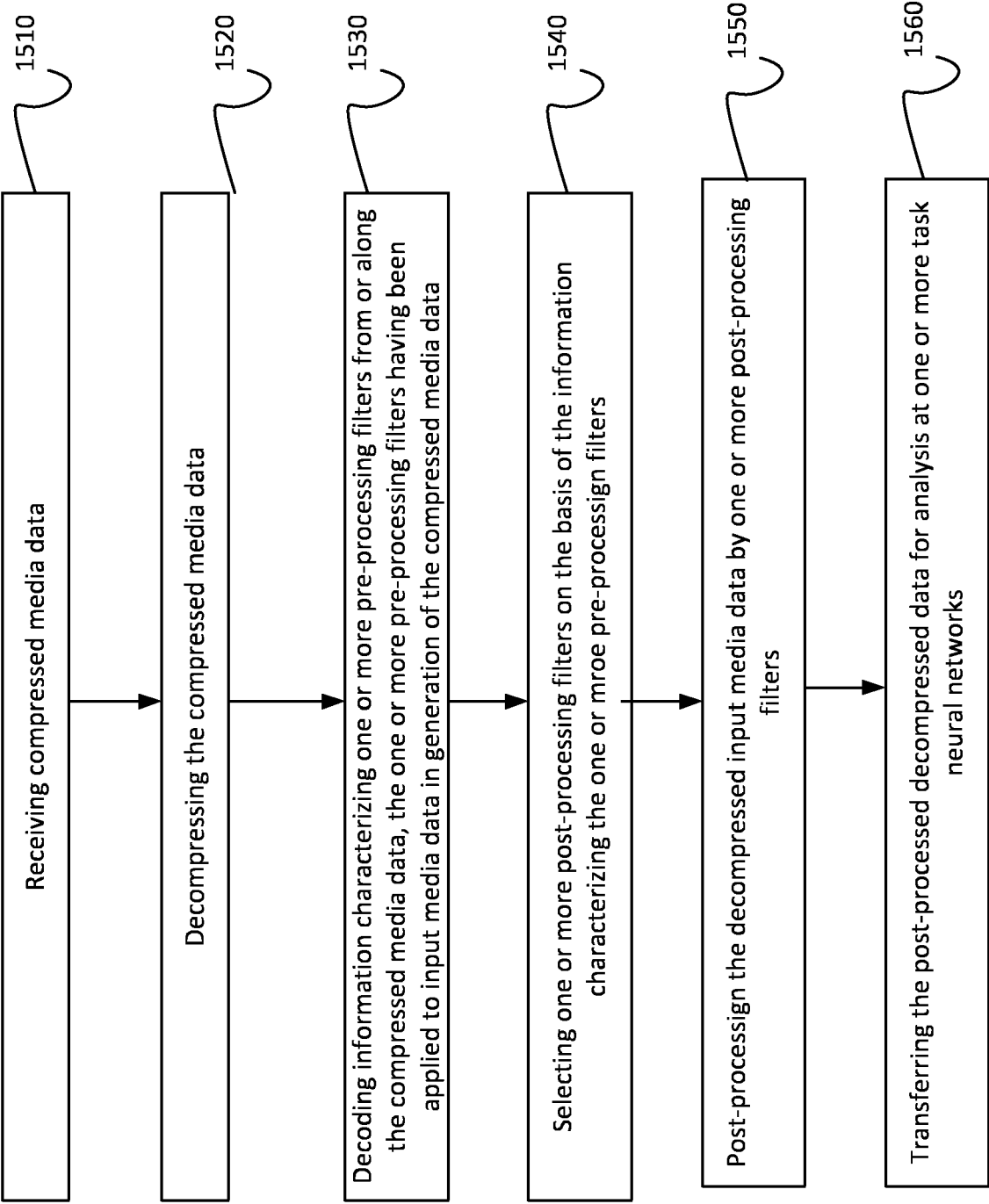


Fig. 15

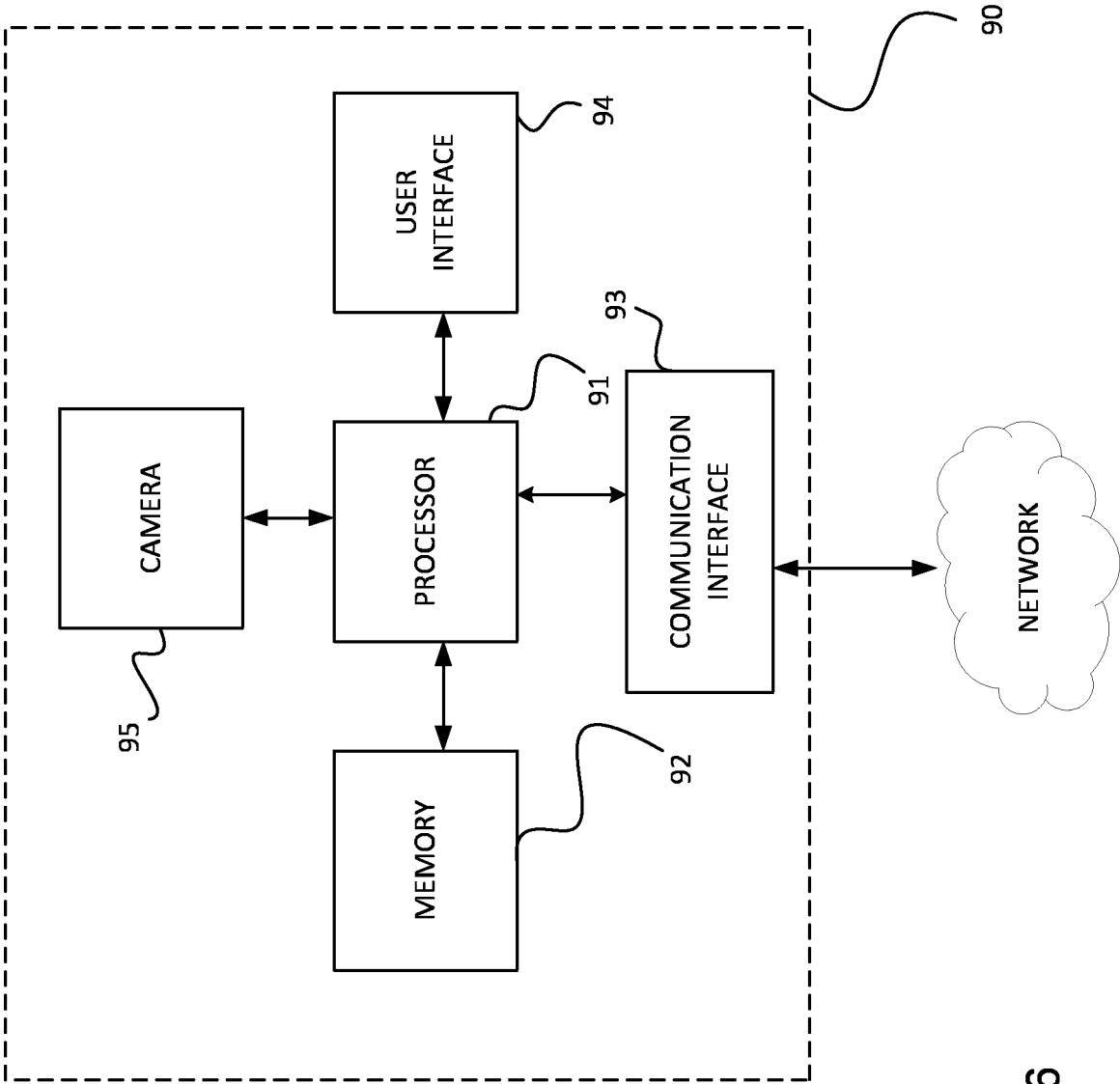


Fig. 16

INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2022/050663

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06N, G06V, H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
FI, SE, NO, DK

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

EPODOC, EPO-Internal full-text databases, including translations from Asian languages, WPIAP, BIOSIS, COMPDX, EMBASE, INSPEC, MEDLINE, NPL, TDB, XP3GPP, XPAIP, XPESP, XPETSI, XPI3E, XPI3ES, XPIEE, XPIETF, XPIOP, XPIPCOM, XPJPEG, XPMISC, XPOAC, XPRD, XPTK, XPSRNG, IPRally, MPEG

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 10681361 B2 (WANG ZEHAN [GB] et al.) 09 June 2020 (09.06.2020) Col. 60, ll. 19-59; Col. 66, ll. 3-21; Col. 66, ll. 55-62; Col. 69, ll. 38-47; Claims 1-2; Figs. 10, 13, 15, 22	1-17
X	US 2021021866 A1 (DJOKOVIC DJORDJE [GB] et al.) 21 January 2021 (21.01.2021) Pars. [0018], [0028], [0068]-[0070], [0089]; Claims 1-2, 13, 15-16; Fig. 1	1, 8, 14-17
A	ISO/IEC JTC1/SC29/WG2. Use cases and requirements for Video Coding for Machines. MPEG document w20705 v1. JTC1/SC29/WG2 N00103. MPEG Document Management System. ISO/IEC, [online], 2021-08-19, XP030297558, [retrieved on 2023-04-12]. Retrieved from <https://dms.mpeg.expert/> The entire document, especially Sec. 2, cf. "Object masking"; Secs. 3.1, A.1, A.2	1-17

☒ Further documents are listed in the continuation of Box C.
 ☒ See patent family annex.

* Special categories of cited documents:	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"D" document cited by the applicant in the international application	"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
"E" earlier application or patent but published on or after the international filing date	"&" document member of the same patent family
"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	

Date of the actual completion of the international search 18 April 2023 (18.04.2023)	Date of mailing of the international search report 18 April 2023 (18.04.2023)
Name and mailing address of the ISA/ FI Finnish Patent and Registration Office FI-00091 PRH, FINLAND Facsimile No. +358 29 509 5328	Authorized officer Teemu Elo Telephone No. +358 29 509 5000

INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2022/050663

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	LEE, Y. et al. [VCM] Response to CfE: Object detection results with the FLIR dataset. m56572. MPEG Document Management System ISO/IEC JTC1/SC29/WG2, [online], 2021-04-21, XP030295066, [retrieved on 2023-03-20]. Retrieved from < https://dms.mpeg.expert/ > The entire document	1-17
A	DUAN, L. et al. Video Coding for Machines: A Paradigm of Collaborative Compression and Intelligent Analytics. In: IEEE Transactions on Image Processing IEEE, 2020-08-28, Vol. 29, pp. 8680-8695, XP011807613, <DOI:10.1109/TIP.2020.3016485> The entire document	1-17

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2022/050663

US 10681361 B2	09/06/2020	GB 201502753 D0	08/04/2015
		GB 201617482 D0	30/11/2016
		GB 201617471 D0	30/11/2016
		EP 3298575 A1	28/03/2018
		US 2018130178 A1	10/05/2018
		US 10499069 B2	03/12/2019
		GB 201519425 D0	16/12/2015
		US 11234006 B2	25/01/2022
		EP 3278559 B1	05/05/2021
		US 2018122048 A1	03/05/2018
		US 2017345130 A1	30/11/2017
		GB 2539846 B	01/11/2017
		US 10692185 B2	23/06/2020
		GB 2543958 B	27/09/2017
		GB 2539846 A	28/12/2016
		EP 3259919 A1	27/12/2017
		US 10516890 B2	24/12/2019
		US 2018130180 A1	10/05/2018
		WO 2016132154 A1	25/08/2016
		US 10666962 B2	26/05/2020
		GB 2539845 B	12/07/2017
		GB 2540889 A	01/02/2017
		GB 201617481 D0	30/11/2016
		EP 3259920 A1	27/12/2017
		EP 3259912 A1	27/12/2017
		EP 3259918 A1	27/12/2017
		WO 2016132149 A1	25/08/2016
		DE 202016009091 U1	10/02/2022
		WO 2016132153 A1	25/08/2016
		GB 2539845 A	28/12/2016
		US 10825138 B2	03/11/2020
		WO 2016132151 A1	25/08/2016
		US 2018129918 A1	10/05/2018
		EP 3259916 A1	27/12/2017
		US 10547858 B2	28/01/2020
		GB 201604672 D0	04/05/2016
		EP 3259912 B1	01/09/2021
		WO 2016132146 A1	25/08/2016
		WO 2017144881 A1	31/08/2017
		WO 2016132145 A1	25/08/2016
		GB 201710000 D0	09/08/2017
		WO 2016132150 A1	25/08/2016
		DE 202016009100 U1	05/04/2022
		US 2018122047 A1	03/05/2018
		GB 2543958 A	03/05/2017
		WO 2016156864 A1	06/10/2016

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2022/050663

EP 3493149 A1	05/06/2019
WO 2016132148 A1	25/08/2016
US 10630996 B2	21/04/2020
GB 2548749 B	13/12/2017
US 10904541 B2	26/01/2021
WO 2016132147 A1	25/08/2016
DE 202016009107 U1	27/04/2022
GB 201505544 D0	13/05/2015
GB 201503427 D0	15/04/2015
US 2017347061 A1	30/11/2017
GB 201617478 D0	30/11/2016
US 2018131953 A1	10/05/2018
DE 202016009102 U1	22/04/2022
GB 201508742 D0	01/07/2015
GB 201617475 D0	30/11/2016
US 2017347110 A1	30/11/2017
EP 3259915 A1	27/12/2017
EP 3259911 B1	07/04/2021
WO 2017158363 A1	21/09/2017
GB 2543429 A	19/04/2017
GB 201519687 D0	23/12/2015
ES 2873845 T3	04/11/2021
ES 2872724 T3	02/11/2021
US 10582205 B2	03/03/2020
EP 3259913 A1	27/12/2017
US 2017347060 A1	30/11/2017
US 10681361 B2	09/06/2020
WO 2017158341 A1	21/09/2017
GB 2548749 A	27/09/2017
US 2018130177 A1	10/05/2018
EP 3278559 A1	07/02/2018
US 10887613 B2	05/01/2021
EP 3259913 B1	12/05/2021
DE 202016009089 U1	10/02/2022
EP 3298776 A1	28/03/2018
EP 3298576 A1	28/03/2018
GB 201603144 D0	06/04/2016
EP 3259916 B1	26/05/2021
GB 2543429 B	27/09/2017
GB 201604345 D0	27/04/2016
US 2018130179 A1	10/05/2018
GB 201507141 D0	10/06/2015
DE 202016009101 U1	05/04/2022
EP 3298576 B1	23/12/2020
EP 3259910 A1	27/12/2017
EP 3259911 A1	27/12/2017
GB 2540889 B	12/07/2017

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2022/050663

WO 2016132152 A1	25/08/2016
US 11528492 B2	13/12/2022
GB 201511231 D0	12/08/2015
EP 3259910 B1	28/04/2021
EP 3259918 B1	19/08/2020
US 10623756 B2	14/04/2020
EP 3298575 B1	12/02/2020
EP 3259914 A1	27/12/2017
US 10523955 B2	31/12/2019
US 2020280730 A1	03/09/2020
US 2017374374 A1	28/12/2017

US 2021021866 A1	21/01/2021	WO 2019220095 A1	21/11/2019
		US 11582481 B2	14/02/2023
		EP 3794828 A1	24/03/2021

CLASSIFICATION OF SUBJECT MATTER

IPC

H04N 19/117 (2014.01)**H04N 19/85** (2014.01)**G06N 3/04** (2023.01)**G06N 3/08** (2023.01)**G06V 10/20** (2022.01)**G06T 9/00** (2006.01)