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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
09 January 2020 (09.01.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/008104 A1

(51) International Patent Classification:
G06T 9/00 (2006.01) *G06N 3/04* (2006.01)
H04N 19/192 (2014.01) *H04N 19/176* (2014.01)
H04N 19/196 (2014.01) *G06N 20/00* (2019.01)

(21) International Application Number:
PCT/FI2019/050483

(22) International Filing Date:
20 June 2019 (20.06.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20185611 02 July 2018 (02.07.2018) FI

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

(72) Inventors: **CRICRI, Francesco**; Massunkuja 1 A 14, 33100 Tampere (FI). **AYTEKIN, Caglar**; Riihuhdankatu 13 B 6, 33580 Tampere (FI).

(74) Agent: **NOKIA TECHNOLOGIES OY** et al.; Ari Aarnio, 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, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

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

(54) **Title:** A METHOD, AN APPARATUS AND A COMPUTER PROGRAM PRODUCT FOR IMAGE COMPRESSION

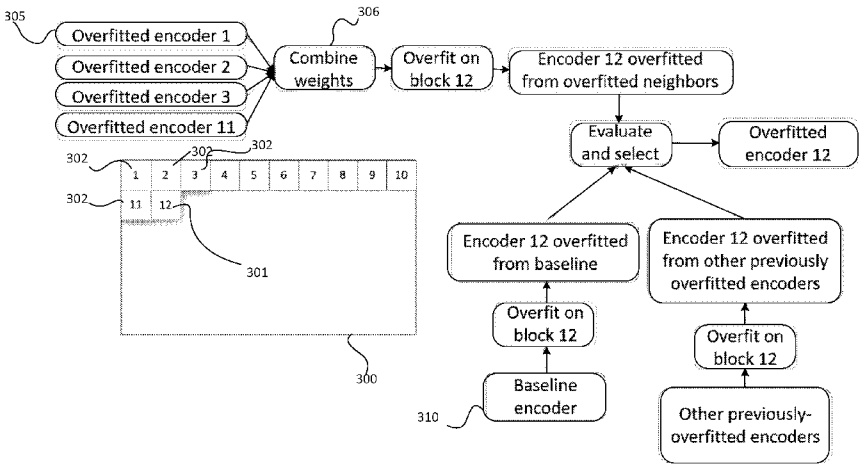

Fig. 3

(57) **Abstract:** The embodiments relate to a method and to a technical equipment implementing the method for image, video, or audio encoding or decoding. The method comprises receiving input data divided into a plurality of blocks; overfitting a first neural encoder network for a first block of the data based on a baseline encoder network (310); encoding the first block by the first overfitted neural encoder network (305); overfitting a second neural encoder network for at least one subsequent block of the data based on a combination of neural encoder networks used for previous blocks (306) and/or the baseline encoder network; and encoding the at least one subsequent block by the second overfitted neural encoder network. Additionally, a block residual defining a difference between an original block of data and a decoded block of the data may be received on the decoder side.

WO 2020/008104 A1

A METHOD, AN APPARATUS AND A COMPUTER PROGRAM PRODUCT FOR IMAGE COMPRESSION

Technical Field

The present solution generally relates to an image or video compression. In particular, the solution relates to neural image (or video) compression.

Background

Many practical applications rely on the availability of semantic information about the content of the media, such as images, videos, etc. Semantic information is represented by metadata which may express the type of scene, the occurrence of a specific action/activity, the presence of a specific object, etc. Such semantic information can be obtained by analyzing the media.

The analysis of media is a fundamental problem which has not yet been completely solved. This is especially true when considering the extraction of high-level semantics, such as object detection and recognition, scene classification (e.g., sport type classification) action/activity recognition, etc.

The development of various neural network techniques has enabled learning to recognize image content directly from the raw image data, whereas previous techniques consisted of learning to recognize image content by comparing the content against manually trained image features. Very recently, neural networks have been adapted to take advantage of visual spatial attention, i.e. the manner how humans conceive a new environment by focusing first to a limited spatial region of the scene for a short moment and then repeating this for a few more spatial regions in the scene in order to obtain an understanding of the semantics in the scene.

Summary

Now there has been invented an improved method and technical equipment implementing the method. Various aspects of the invention 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 of the invention are disclosed in the dependent claims.

According to a first aspect, there is provided a method comprising receiving input data divided into a plurality of blocks; overfitting a first neural encoder network for a first block of the data based on a baseline encoder network; encoding the first block by the first overfitted neural encoder network; overfitting a second neural encoder network for at least one subsequent block of the data based on a combination of neural encoder networks used for previous blocks and/or the baseline encoder network; and encoding the at least one subsequent block by the second overfitted neural encoder network.

According to a second aspect, there is provided a method for a neural decoder network comprising receiving a block residual defining a difference between an original block of data and a decoded block of the data; based on the residual, recovering the original block to be used as ground-truth data; and overfitting the neural decoder network based on the ground-truth data.

According to a third 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 receive input data divided into a plurality of blocks; overfit a first neural encoder network for a first block of the data based on a baseline encoder network; encode the first block by the first overfitted neural encoder network; overfit a second neural encoder network for at least one subsequent block of data based on a combination of neural networks used for previous blocks and/or the baseline encoder network; and encode the at least one subsequent block by the second overfitted neural encoder network.

According to an embodiment, the apparatus further configured to determine which one of the overfitted neural encoder networks performs the best; and select such overfitted neural encoder network for a current block.

According to an embodiment, the performance is determined according to one or both of the following aspects: a reconstruction quality or a bitrate.

According to an embodiment, the data comprises image data, video data, or audio data.

According to a fourth 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 receive a block residual defining a difference between an original block of data and a decoded block of the data; based on the residual, recover the original block to be used as ground-truth data; and overfit the neural decoder network based on the ground-truth data.

According to an embodiment, the apparatus is further being configured to receive a weight residual from a transmitter, the weight residual defining the difference between weights of the decoder before and after an overfitting.

According to an embodiment, the data comprises image data, video data, or audio data.

According to a fifth aspect, there is provided a 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 data divided into a plurality of blocks; overfit a first neural encoder network for a first block of the data based on a baseline encoder network; encode the first block by the first overfitted neural encoder network; overfit a second neural encoder network for at least one subsequent block of data based on a combination of neural networks used for previous blocks and/or the baseline encoder network; and encode the at least one subsequent block by the second overfitted neural encoder network.

According to a sixth aspect, there is provided a computer program product comprising computer program code configured to, when executed on at least one processor, cause an apparatus or a system to receive a block residual defining a difference between an original block of data and a decoded block of the data; based on the residual, recover the original block to be used as ground-truth data; and overfit the neural decoder network based on the ground-truth data.

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

Description of the Drawings

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

Fig. 1 shows an example of a computer system according to an embodiment;

Fig. 2 shows an embodiment for a training process of a neural auto-encoder;

Fig. 3 shows a sequential encoder overfitting according to a first embodiment;

Fig. 4 shows another example of a sequential encoder overfitting according to a first embodiment;

Fig. 5 shows a decoder overfitting according to a second embodiment;

Fig. 6 is a flowchart illustrating a method according to an embodiment; and

Fig. 7 is a flowchart illustrating a method according to another embodiment.

Description of Example Embodiments

In the following, several embodiments will be described in the context of image compression. In particular, the several embodiments enable using neural network for image compression/decompression. It is to be noted, however, that the embodiments are not limited to compression/decompression of images, but compression/decompression of video as well. Therefore, any time term “image” is used in the following description, it is appreciated that the term covers also “video” or “video frame”. In addition, the present embodiments are also applicable with other media content, such as audio, speech, etc. For example, the data block corresponding of the image block concept in audio signals may be an audio frame. Spatially neighboring image blocks may correspond to temporally neighboring audio frames. In addition, similar concepts of image blocks can be used when considering audio spectrogram images.

Figure 1 shows a computer system suitable to be used in data processing. The generalized structure of the computer system will be explained in accordance with the functional blocks of the system. Several functionalities can be carried out with a single physical device, e.g. all calculation procedures can be performed in a single processor if desired. A data processing system of an apparatus according to an example of Fig. 1 comprises a main processing unit 100, a memory 102, a storage device 104, an input device 106, an output device 108, and a graphics subsystem 110, which are all connected to each other via a data bus 112.

The main processing unit 100 is a processing unit comprising processor circuitry and arranged to process data within the data processing system. The memory 102, the storage device 104, the input device 106, and the output device 108 may include conventional components as recognized by those skilled in the art. The memory 102 and storage device 104 store data within the data processing system 100. Computer program code resides in the memory 102 for implementing, for example, computer vision process. The input device 106 inputs data into the system while the output device 108 receives data from the data processing system and forwards the data, for example to a display, a data transmitter, or other output device. The data bus 112 is a conventional data bus and while shown as a single line it may be any combination of the following: a processor bus, a PCI bus, a graphical bus, an ISA bus. Accordingly, a skilled person readily recognizes that the apparatus may be any data processing device, such as a computer device, a personal computer, a server computer, a mobile phone, a smart phone or an Internet access device, for example Internet tablet computer.

It needs to be understood that different embodiments allow different parts to be carried out in different elements. For example, various processes of the data processing system may be carried out in one or more processing devices; for example, entirely in one computer device, or in one server device or across multiple user devices. The elements of data processing may be implemented as a software component residing on one device or distributed across several devices, as mentioned above, for example so that the devices form a so-called cloud.

A neural network (NN) is a computation graph comprising several layers of computation. Each layer comprises 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 a weight. The weight may be used for scaling the signal passing through the associated connection. Weights are usually 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 examples of architecture 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 preceding layers and provides its output as the input for one or more of the subsequent layers. Also, units inside certain layers take input from units in one or more of preceding layers, and provide output to one or more of the following layers.

Initial layers (those close to the input data) extract semantically low-level features such as edges and texture 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 networks, there is a feedback loop, so that the network becomes stateful, i.e., it is able to memorize information or a state.

Neural networks may be 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 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 changes 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 may happen 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 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.

In the present description terms "neural network", "neural net" and "network" are used interchangeably, and also the "weights" of neural network may be referred to as "learnable parameters" or "parameters".

A neural network has two main modes of operation: training phase and testing phase. The training phase is the development phase, where the network learns to perform the final task. Training may involve iteratively updating the weights between units. Training a neural network is an optimization process, where the goal of optimization or training process is to make the neural network to 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 the previously unseen data, i.e., data which was not used for training the neural network. This may be 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 neural network. In particular, the errors on the training set and on the validation set are monitored during the training process to understand the following things:

- If the neural network is learning at all – in this case, the training set error should decrease, otherwise the neural network is in the region of underfitting.
- If the network is learning to generalize – in this case, also the validation set error needs to decrease and not to be 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 neural network is in the regime of overfitting (i.e. optimization). This means that the neural network 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.

Recently, neural image compression and decompression systems are based on neural auto-encoders, or simply auto-encoders. An auto-encoder may comprise two neural networks, one of which is the neural encoder (also referred to as "encoder" in this description for simplicity) and the other is the neural decoder (also referred to as "decoder" in this description for simplicity). The encoder is configured to map the input data (such as an image, for example) to a representation which is more easily or more efficiently compressed. The decoder gets the compressed version of the data and is configured to de-compress it, thus reconstructing the data.

The two networks in the auto-encoder may be trained simultaneously, in an end-to-end fashion. The training may be performed by using at least a reconstruction loss, which trains the auto-encoder to reconstruct the image correctly. An example of reconstruction loss is the mean squared error (MSE). In order to achieve also high compression gains, an additional loss on the output of the encoder may be used.

Fig. 2 illustrates an example of a neural auto-encoder training process for image compression.

After training, the output of the encoder may be binarized and entropy-coded. Binarization is a non-differentiable operation, so it cannot be used during training of the encoder, because it is not possible to obtain useful gradients for the encoder. However, even if there is a binarization operation, it is still possible to train the decoder. One common training strategy is to have two alternate training steps: in one training step no binarization is used (optionally, a differentiable approximation of the binarization may be used instead), and both encoder and decoder are trained; in the second training step, binarization is used and only the decoder is trained.

In machine learning and neural networks, the training may be performed on a big dataset, which is good representative of the data that may be used at test time. This way, the network is trained to generalize on unseen data (but which is still sufficiently similar to the training data). However, the performance of a neural network can drastically be improved if the network is optimized on the target input data on which it will be used. In the present disclosure this specific optimization to one or more test input data is referred to as "overfitting" or "fine-tuning". Overfitting may refer to optimizing or training (e.g., updating the learnable parameters of) a neural network on a certain test datum or several test data, as opposed to optimizing on a general set of training data. The test data is the data on which the neural network is applied

when it is utilized for its purpose (for example, the test data may include an image that is to be compressed). Overfitting is for example beneficial when there is a sudden domain shift in the data, and especially if the domain shift happens continuously and gradually. A data domain shift means that the data domain or context or type changes, for example a camera may start to capture data from a different-looking scene, where the difference may be in the lighting, in the type or amount of objects, in the type or amount of motion, in the type or amount of texture, etc. A neural network which is trained on a different data domain than the one on which it is run may perform sub-optimally.

The present embodiments relate to neural image (or video) compression. This may include using neural networks for compressing and/or de-compressing images (or other data), with high compression gain and a high reconstruction quality. The compression gain can be measured by the number of bits of the encoded or compressed representation. The reconstruction quality can be measured by a certain metric which compares the original image and the de-compressed or reconstructed image.

The various embodiments provide a set of techniques for optimizing image compression auto-encoders on the specific input data ("overfitting"), and in a way which improves the encoding speed and the reconstruction quality, or alternatively improves the encoding speed and the compression gain.

In the various embodiments of the solution, an auto-encoder is used as an example. The auto-encoder is optimized (i.e., overfitted) to a specific input data on which it is used. This optimization may be performed at network utilization time, not at training time.

According to a first embodiment of the present solution, the encoding is sped up via sequential neural encoder network overfitting. According to a second embodiment, the decoding quality is improved via neural decoder network overfitting.

In general, the neural decoder network should not be optimized for a certain input data because sending the data to the decoder for performing the optimization may require too many bits. However, in the second embodiment, a strategy allowing trade-offs between bitrates and decoding quality is introduced.

Overfitting (i.e. optimization) may be more effective if it is performed on an image-block level. So, there will be as many optimized networks as there are blocks in the image.

A baseline network is assumed to be available, which has been trained on a large dataset. This baseline network is overfitted to the first block of the image. For subsequent blocks, the overfitting is performed by starting from the neural network overfitted on the neighboring blocks. This will speed up the overfitting process as it may require much less training iterations than if started from the baseline network.

However, it is possible that the baseline network is better to act as a starting neural network in some situations, thus multiple evaluation strategies are introduced for selecting the starting neural network.

An initial network to be overfitted can be one of the already overfitted ones for neighboring blocks, without combining their weights. So, each of the previously-overfitted networks may be overfitted on the current block and then evaluated to determine the best performing network. According to an embodiment, for at least one of the previously-overfitted networks, the multiple weight versions obtained at different training iterations during the previous overfitting process may be stored (e.g., every 100 iterations). Each of such weight versions may be referred to as an intermediate version of a neural network. Then, overfitting on the current block may be done on a plurality of network versions previously-overfitted on the at least one neighboring block. Finally, a comparison of the plurality of overfitted networks may be performed. This comparison may include the network(s) overfitted from baseline, the network(s) overfitted from previously-overfitted network(s) on neighboring blocks, the network(s) overfitted from intermediate version(s) of previously-overfitted networks(s) on neighboring blocks, and network(s) overfitted from a combination of previously-overfitted on neighboring blocks

Regarding the second embodiment, the neural decoder network overfitting process may be performed at the encoder side, where the neural decoder network is overfitted to one block of every N blocks. If this overfitting results into a much improved PSNR (Peak Signal-to-Noise Ratio) for the considered block and for its neighbors compared to the bitrate increase caused by sending the necessary additional data to the decoder's side, then also the decoder's side uses the overfitted neural network.

In an additional embodiment, if there are multiple auto-encoders in order to achieve variable bitrate, neural decoder network overfitting may allow using a lower-dimensional auto-encoder so that the PSNR is similar to the one obtained by a higher-dimensional auto-encoder but with a lower bitrate, thus resulting into higher compression efficiency.

In the example of various embodiments, there is one entity preforming the encoding or compression of a data, here referred to as the "transmitter", and one entity performing the decoding or de-compression of the encoded data, here referred to as the "receiver". Thus, the transmitter may comprise the neural encoder network, and the receiver may comprise the neural decoder network. Although terms "transmitter" and "receiver" are used, it should be noticed that the solution is not limited only to applications where encoded data is transmitted from the transmitter to the receiver. For example, the encoded data may be saved on the same device from which is will be decoded, or it may be saved on another device which will be used for moving the data to another memory device from which the decoder will decoded the encoded data.

In various embodiments, the auto-encoder has been trained on a large collection of images, such as ImageNet, Places 2, or similar datasets. This pre-trained auto-encoder is referred in this disclosure to as "baseline auto-encoder" or "baseline network" or simply as "baseline". The encoder and decoder of the autoencoder, will be referred to as the "baseline encoder" and "baseline decoder" respectively, or simply as "baselines", when it is clear that both encoder and decoder are being referred to. The training of the baseline network is assumed to have been performed for the task of image (or other specific data) compression, thus by using at least one reconstruction loss and at least one compression loss. The baseline network will be the starting point for overfitting either the encoder or the decoder or both. Here, the baseline network is considered to be a neural network which is able to compress and decompress well any piece of data which is not too different from the training data. Thus, it is a neural network which is able to generalize. Although having such generalization characteristics is beneficial, the present embodiments are not restricted to this requirement.

As mentioned, neural network can be optimized (i.e. "overfitted" or "fine-tuned") on a specific input data. Such optimization operation is a training operation, thus

comprising one or more training iterations, where the weights of the neural network are changed in order to improve the performance of the neural network on the input data. In this operation, the neural network may deviate from its generalization capabilities and will become instead specific or fine-tuned on the input data on which it was optimized. This optimization operation may performed at inference time and not at training time. I.e., it may be performed during encoding or decoding the data.

The overfitting may be performed on a block-level. Therefore, an image can be divided into blocks and a neural network will be fine-tuned on at least one block. According to an embodiment the blocks are non-overlapping. For video data, a network may be optimized on at least one block and at least one frame, or alternatively it may be optimized on at least one whole frame.

The first and second embodiments are described in more detailed manner in the following.

First embodiment: sequential encoder overfitting

In an example of the first embodiment, shown in Fig. 3 evaluation is performed by overfitting the candidate neural networks on the current block, and the choosing the best neural network.

The transmitter receives an input image 300 (or other data) which needs to be encoded. The transmitter has at least one baseline encoder network 310 (i.e. encoder of the pretrained auto-encoder). The transmitter may divide the image 300 into blocks (1, 2, 3, 4, …., 12, …), if the image hasn't been already divided. After that, the transmitter overfits a neural encoder network 305 to each block of the image. Alternatively, the transmitter may decide to overfit neural encoder network to a subset of all blocks, and to use the neural network overfitted on neighboring blocks for encoding the blocks on which no overfitting was performed. However, for the sake of simplicity, in an example of this embodiment, the transmitter has overfitted one neural encoder network for each block.

For overfitting on the first block (which may be anywhere in the image but it is considered here to be the top-left most block with number 1), the only available neural encoder network is the baseline encoder 310, so the overfitting will start from the baseline encoder 310. A copy of the original baseline encoder may be made

and kept at the transmitter side. In general, the overfitting of the first block starts from a baseline encoder. In one embodiment, the baseline encoder may be determined to be an encoder corresponding to a block of a previous image or video frame having same or nearby (e.g. adjacent) location with respect to the first block. The baseline encoder may be also determined based on similarity of the first block of the current image or video frame and a block of a previous image or video frame.

A neural network is defined by its topology or architecture (e.g., number and type of layers), and by its weights. It is assumed in this example that only the weights of the neural encoder network are changed during the optimization processes, but the topology may be changed as well, for example based on type of content in the relevant block of image. The neural network can be characterized or represented by a point in the weight space, where each dimension of this space is a weight of the neural network. The baseline network may be considered to be a point in weight space which is relatively close to the optimal points for all images, but not too close to any of those optimal points. By the optimization operation, the optimized neural network gets closer in weight space to the optimal neural network for the data on which it was optimized.

In images, video, audio, speech, text and other types of data, there is high structural correlation which means that there is lot of either spatial or temporal correlations, or both. This correlation means that two nearby blocks are likely to have similar content, and thus two neural networks optimized separately on these two blocks are likely to be close in weight space, or anyway closer than their distance from the baseline. Thus, optimizing for a second block (and subsequent blocks) by starting from the neural network optimized on the first block (or neighboring blocks) is likely to require much less training iterations to converge. Due to this the encoding speed may drastically increase.

If multiple neighboring blocks 302 (with respect to the current block 301) were used for overfitting encoders 305, the overfitting for the current block 301 can be started from a combination of the overfitted neural networks on those neighboring blocks 302. The combination may be an average of the weights 306, or any suitable neural network combining method.

In some situations, such as at objects boundaries, it may happen that the current block's 301 neighboring blocks are very dissimilar from the current block 301, so the

neural networks overfitted to those neighboring blocks may be even farther away in weight space than the baseline mode or previously overfitted neural networks. Thus, the transmitter may apply an evaluation phase where the best neural network for the start is selected. This can be made for example by running parallel overfitting sessions from different neural networks used as a start, such as from the "neighboring overfitted neural networks" (the neural networks overfitted on neighboring blocks), from the baseline network, and optionally from any other previously-overfitted neural network. After the multiple overfitting sessions are completed, the neural network performing best on the current block is selected, for example, according to the reconstruction quality, according to the bitrate, or according to a combination of these aspects. Determining which neural network performs the best may include comparing the network overfitted from a combination of networks overfitted on neighboring blocks to the network overfitted from the baseline network

Fig. 4 shows an alternative strategy for the first embodiment, wherein the evaluation is performed by running candidate starting neural networks on the current block 401, and the best performing neural network is then used as the starting neural network for being overfitted to the current block 401. In this example, selecting the best neural network for the current block may include running (i.e. only inference stage, instead of overfitting) the candidate neural networks on the current block and of determining which neural network performs best without any optimization on the current block. The best neural network may be selected, and used as a reference encoder to overfit on the current block. The motivation is that if a neural network is already performing well on the current block 401, it is likely to be close in weight space to the optimal point for the current block 401. This strategy has the advantage of avoiding having multiple overfitting session for the current block 401, as only inference phase is run on all candidate neural networks and then only one neural network is optimized.

In a case, where the data type is video, a further additional neural network to be evaluated (either by running the overfitting process or by running the inference step) may be a neural network overfitted on the current block but on the previous frame, or on neighboring blocks on the previous frame, or a combination thereof. Furthermore, neural networks from multiple previous frames may be considered too.

After the encoder has been overfit to a certain input block, the original decoder may still be used at inference stage because during overfitting the original decoder was used and its weights were not modified. This enables optimizing the encoder during data transmission without a need to send updated weights of the decoder to the receiver.

Second embodiment: decoder overfitting

Overfitting the decoder means that the neural decoder network is further optimized on the current block so that the decoding or reconstruction of the encoded block is improved. Fig. 5 illustrates an example of decoder overfitting. The transmitter 510 overfits the decoder 512 by using the original block as ground-truth, in order to obtain the overfitted decoder 514. The overfitted decoder 514 will be evaluated and compared to other decoders which are available at receiver's 520 side, in order to decide if it is worth using such overfitted decoder 514 with respect to the bitrate increase of sending the needed additional information. This evaluation is done for the current block and N subsequent blocks. If it is worth, in order to enable the receiver 520 to perform the overfitting, the transmitter 510 computes the block residual and sends it to the receiver 520. The receiver 520 uses the block residual to recover the high quality block and uses it as ground-truth for performing the overfitting. It is realized that the overfitting is done at the receiver (decoder) device based on the residual, instead of just receiving the overfitted version of the neural network from the transmitter. This avoids sending the updated decoder weights to receiver 520 and reduces the overhead to the block residuals.

Training a neural network may involve using ground-truth data in order to compute the loss or error value, which is then differentiated with respect to the network's weights, and the obtained gradients are used for updating the weights' value. The ground-truth for training a decoder is the desired reconstructed blocks, which usually are the original blocks which are input to the encoder. However, the original blocks are available only at transmitter's side. Therefore, two possible alternatives are possible:

a) The transmitter sends the block residual to the receiver, i.e., the difference between the original block and the decoded block. This way, the receiver can recover the original block and use it as ground-truth for performing the overfitting. The additional signaling associated with this option may include informing the receiver that the transmitted data is the block

residual for a certain block or a certain image, using unique IDs for both, for example a block identifier and/or an image identifier or a frame number. Fig. 4 illustrates this.

b) The transmitted performs the overfitting of the decoder, and sends to the receiver the decoder's weight residual, i.e., the difference between the weights of the decoder before and after the overfitting. The additional signaling associated with this option may include informing the receiver that the transmitted data is weights' residual and each single weight residual value may be associated to an identifier of the weight to be applied to. In order to reduce the amount of signaling, one may send the weight residual for all weights, where the order of the weights residuals implicitly identify what weights they need to be applied to, and where many weights residuals may be zero. Other suitable ways of associating the weights residuals to the correct weights may be used.

The transmitter may choose to consider both of the above two options initially. Then, it may compute the bitrate increase separately for each option and select the option with minimal bitrate increase. However, in some cases the bitrate increase for allowing the receiver to run an overfitted decoder for the current block may not be worth the reconstruction quality increase. Thus, since subsequent nearby blocks are likely to benefit also by the decoder overfitted to the current block (due to spatial correlation/redundancy in images), the transmitter may take into account the reconstruction quality (e.g., PSNR) increase for the current block and for the subsequent N blocks. If the quality increase for those blocks is worth bitrate increase, then the transmitter may send the additional data (either the block residual or the weights residual) to the receiver.

Also, the transmitter may take into account the baseline and other decoders previously overfitted and which are already available at receiver's side. If the baseline decoder performs well enough (especially compared to the bitrate increase for using an overfitted decoder), the transmitter may not send any additional data. If one of the previously overfitted decoders which are available at receiver's side performs well enough, the transmitter will signal to the receiver to use that overfitted decoder (e.g., by using a unique decoder's ID).

Also, in this embodiment, it is possible that the overfitting starts from the baseline, from the neural networks overfitted on previous blocks, or from neural networks

overfitted on previous frames, or a combination thereof, similarly to what was done in the first embodiment.

As a further embodiment, the overfitted decoder can be used for bitrate saving. The PSNR gain brought up by a better decoder can be exploited for achieving a better compression gain. This means that original PSNR can be achieved with a lower bitrate. This can be implemented for example by having multiple auto-encoders, one for each encoding dimension (e.g., 64 bits and 216 bits). If overfitting the decoders allows to obtain the same or better PSNR with a lower-dimension auto-encoder, and the saved bits are lower than the overhead bits needed by the block residual or by the weights residual, then there is compression gain.

Fig. 6 is a flowchart illustrating a method according to an embodiment. A method comprises receiving 601 input data divided into a plurality of blocks; overfitting 602 a first neural encoder network for a first block of the data based on a baseline encoder network; encoding 603 the first block by the first overfitted neural encoder network; overfitting 604 a second neural encoder network for at least one subsequent block of the data based on a combination of neural networks used for previous blocks and/or the baseline encoder network; and encoding 605 the at least one subsequent block by the second overfitted neural encoder network.

An apparatus according to an embodiment comprises means for receiving input data divided into a plurality of blocks; means for overfitting a first neural encoder network for a first block of the data based on a baseline encoder network; means for encoding the first block by the first overfitted neural encoder network; means for overfitting a second neural encoder network for at least one subsequent block of the data based on a combination of neural encoder networks used for previous blocks and/or the baseline encoder network; and means for encoding the at least one subsequent block by the second overfitted neural encoder network. 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 Fig, 6 according to various embodiments.

Fig. 7 is a flowchart illustrating a method according to another embodiment. A method for neural decoder network comprises receiving 701 a block residual

defining a difference between an original block of data and a decoded block of the data; based on the residual, recovering 702 the original block to be used as ground-truth data; and overfitting 703 the neural decoder network based on the ground-truth data.

An apparatus according to an embodiment comprises means for receiving a block residual defining a difference between an original block of data and a decoded block of the data; based on the residual, means for recovering the original block to be used as ground-truth data; and means for overfitting the neural decoder network based on the ground-truth 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 Fig, 7 according to various embodiments.

The various embodiments may provide advantages. For example, the various embodiments improve the inference speed and decoding quality, or alternatively inference speed and compression efficiency, for neural image (or video) compression.

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 an embodiment. An apparatus may comprise means for performing functions described in the appended claims and throughout the description. The computer program code comprises one or more operational characteristics.

According to an embodiment, said operational characteristics are being defined through configuration by said computer based on the type of said processor, wherein a system is connectable to said processor by a bus, wherein a programmable operational characteristic of the system comprises receiving input

data divided into a plurality of blocks; overfitting a first neural encoder network for a first block of the data based on a baseline encoder network; encoding the first block by the first overfitted neural encoder network; overfitting a second neural encoder network for at least one subsequent block of the data based on a combination of neural networks used for previous blocks and/or the baseline encoder network; and encoding the at least one subsequent block by the second overfitted neural encoder network.

According to another embodiment, the programmable operational characteristic of the system comprises receiving a block residual defining a difference between an original block of data and a decoded block of the data; based on the residual, recovering the original block to be used as ground-truth data; and overfitting the neural decoder network based on the ground-truth data.

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. A method, comprising:
   - receiving input data divided into a plurality of blocks;
   - overfitting a first neural encoder network for a first block of the data based on a baseline encoder network;
   - encoding the first block by the first overfitted neural encoder network;
   - overfitting a second neural encoder network for at least one subsequent block of the data based on a combination of neural encoder networks used for previous blocks and/or the baseline encoder network; and
   - encoding the at least one subsequent block by the second overfitted neural encoder network.

2. The method of claim 1, further comprising:
   - determining which one of the overfitted neural encoder networks performs the best; and
   - selecting such overfitted neural encoder network for a current block.

3. The method according to claim 2, wherein the performance is determined according to one or both of the following aspects: a reconstruction quality or a bitrate.

4. A method for a neural decoder network comprising:
   - receiving a block residual defining a difference between an original block of data and a decoded block of the data;
   - based on the residual, recovering the original block to be used as ground-truth data; and
   - overfitting the neural decoder network based on the ground-truth data.

5. The method according to claim 4, further comprising
   - receiving a weight residual from a transmitter, the weight residual defining the difference between weights of the decoder before and after an overfitting.

6. 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:

- receive input data divided into a plurality of blocks;
- overfit a first neural encoder network for a first block of the data based on a baseline encoder network;
- encode the first block by the first overfitted neural encoder network
- overfit a second neural encoder network for at least one subsequent block of the data based on a combination of neural networks used for previous blocks and/or the baseline encoder network; and
- encode the at least one subsequent block by the second overfitted neural encoder network.

7. The apparatus of claim 6, further being caused to:
   - determine which one of the overfitted neural encoder networks performs the best; and
   - select such overfitted neural encoder network for a current block of the data.

8. The apparatus according to claim 7, wherein the performance is determined according to one or both of the following aspects: a reconstruction quality or a bitrate.

9. The apparatus according to any of the claims 6 to 8, wherein the data comprises image data, video data, or audio data.

10. 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:
   - receive a block residual defining a difference between an original block of data and a decoded block of the data;
   - based on the residual, recover the original block to be used as ground-truth data; and
   - overfit the neural decoder network based on the ground-truth data.

11. The apparatus according to claim 10, further being caused to
   - receive a weight residual from a transmitter, the weight residual defining the difference between weights of the decoder before and after an overfitting.

12. The apparatus according to any of claims 10 or 11, wherein the data comprises image data, video data, or audio data.

13. A 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 data divided into a plurality of blocks;
    - overfit a first neural encoder network for a first block of the data based on a baseline encoder network;
    - encode the first block by the first overfitted neural encoder network
    - overfit a second neural encoder network for at least one subsequent block of the data based on a combination of neural networks used for previous blocks and/or the baseline encoder network; and
    - encode the at least one subsequent block by the second overfitted neural encoder network.

14. A computer program product according to claim 13, wherein the computer program product is embodied on a non-transitory computer readable medium.

15. A computer program product comprising computer program code configured to, when executed on at least one processor, cause an apparatus or a system to:
    – receive a block residual defining a difference between an original block of data and a decoded block of the data;
    – based on the residual, recover the original block to be used as ground-truth data; and
    – overfit the neural decoder network based on the ground-truth data.

16. A computer program product according to claim 15, wherein the computer program product is embodied on a non-transitory computer readable medium.

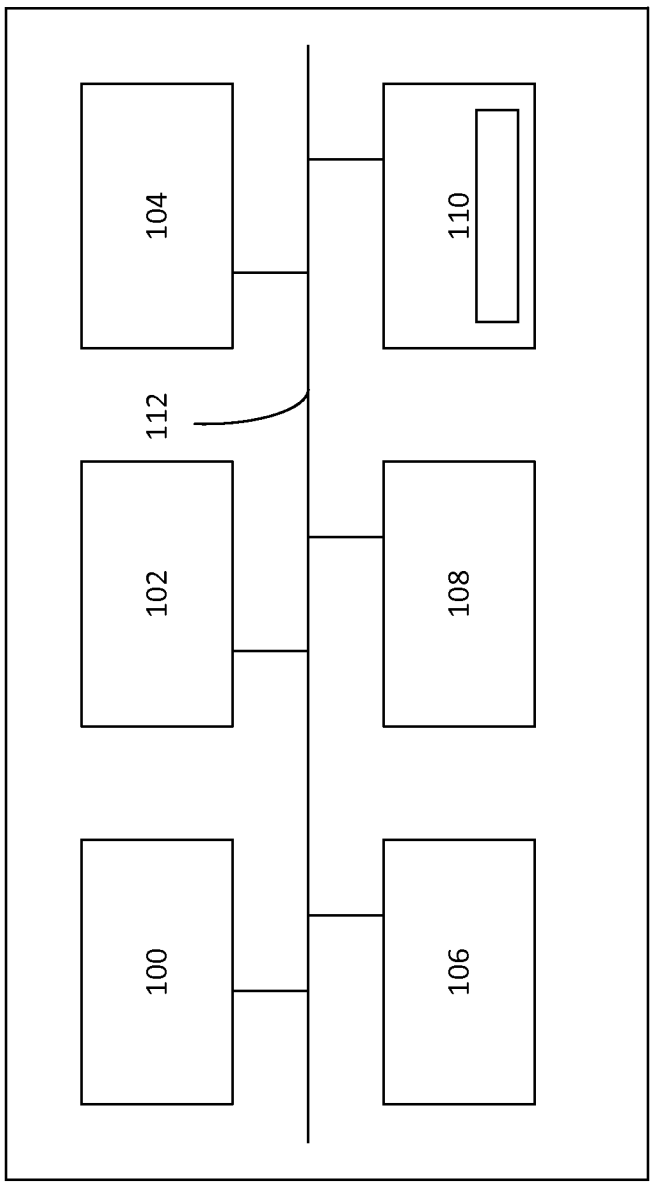

Fig. 1

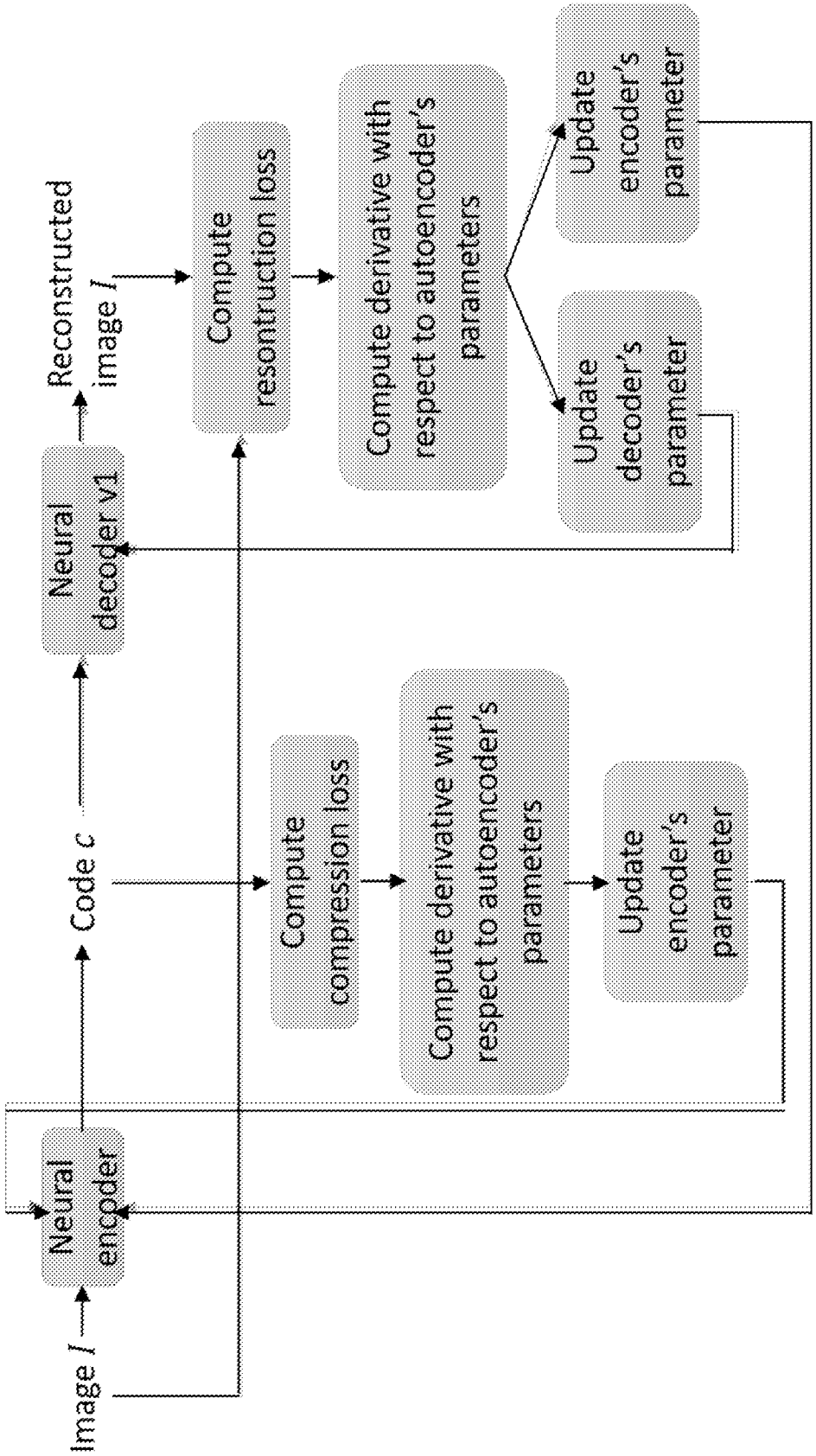
Image I
Neural encoder
Code c
Neural decoder v1
Reconstructed image I'
Compute compression loss
Compute resontruction loss
Compute derivative with respect to autoencoder's parameters
Compute derivative with respect to autoencoder's parameters
Update encoder's parameter
Update decoder's parameter
Update encoder's parameter

Fig. 2

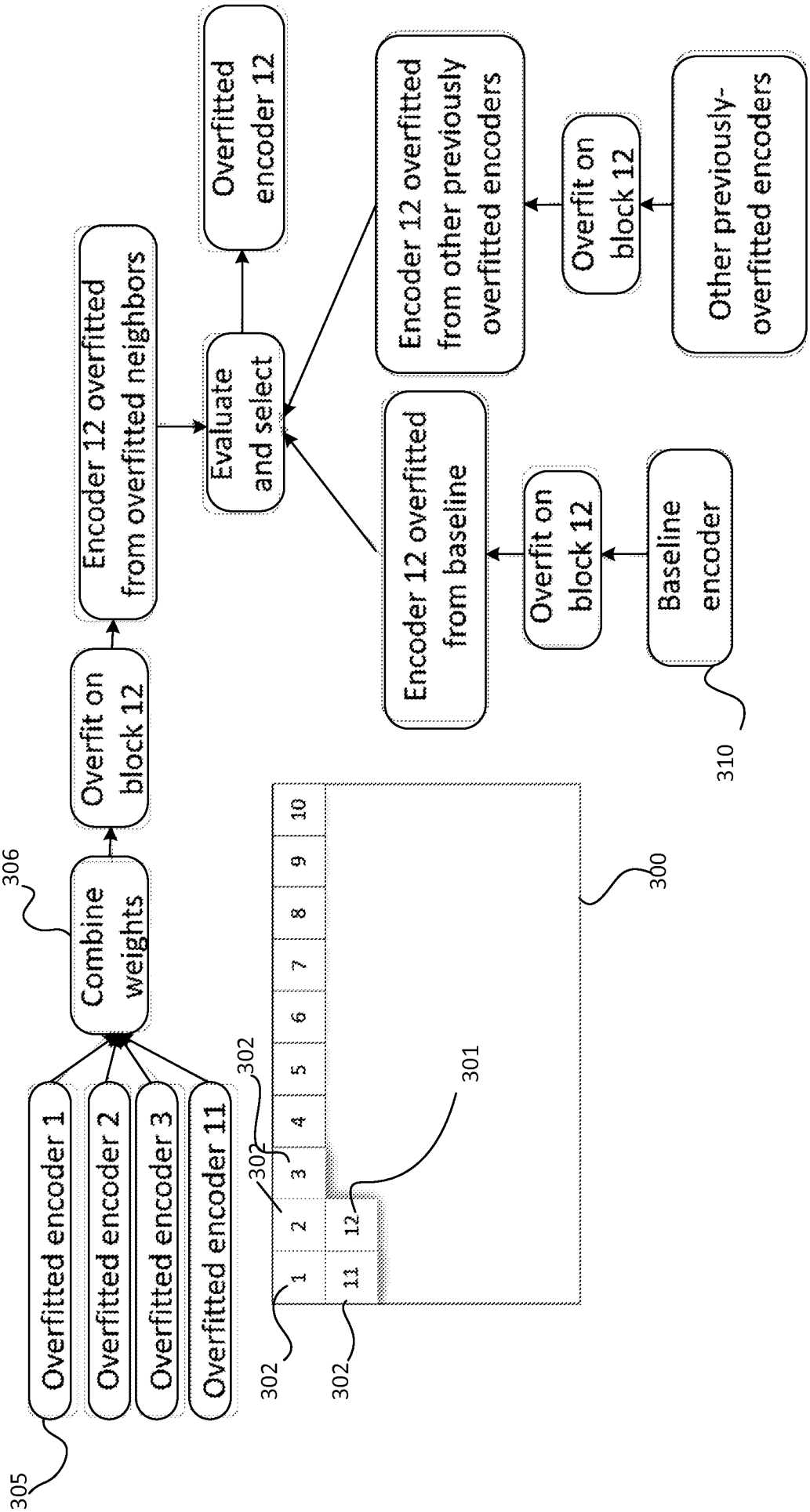
Overfitted encoder 1
Overfitted encoder 2
Overfitted encoder 3
Overfitted encoder 11
305
Combine weights
306
Overfit on block 12
Encoder 12 overfitted from overfitted neighbors
Evaluate and select
Overfitted encoder 12
Encoder 12 overfitted from baseline
Overfit on block 12
Baseline encoder
310
Encoder 12 overfitted from other previously overfitted encoders
Overfit on block 12
Other previously-overfitted encoders
300
301
302
1
2
3
4
5
6
7
8
9
10
11
12

Fig. 3

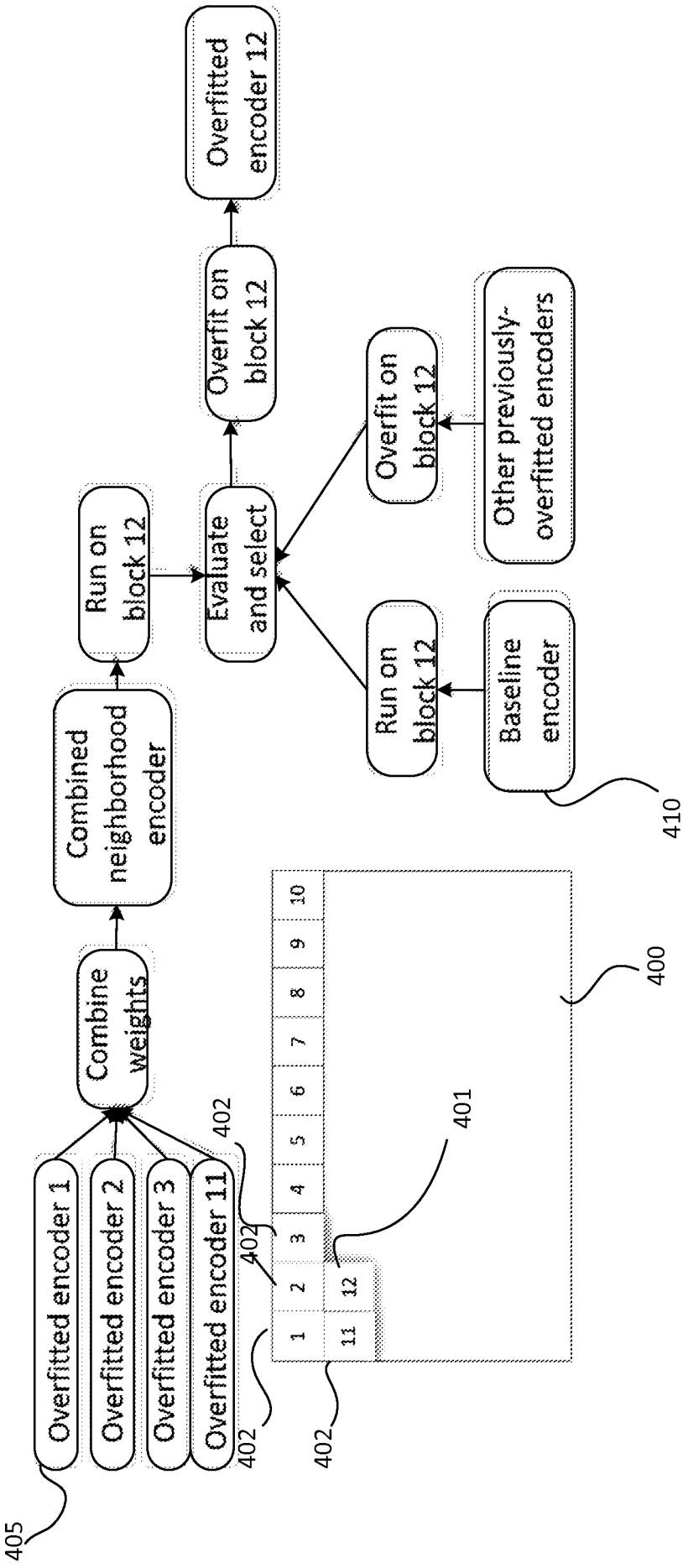
Overfitted encoder 1
Overfitted encoder 2
Overfitted encoder 3
Overfitted encoder 11
Combine weights
Combined neighborhood encoder
Run on block 12
Evaluate and select
Overfit on block 12
Overfitted encoder 12
Run on block 12
Baseline encoder
Overfit on block 12
Other previously-overfitted encoders
1
2
3
4
5
6
7
8
9
10
11
12
400
401
402
402
402
405
410

Fig. 4

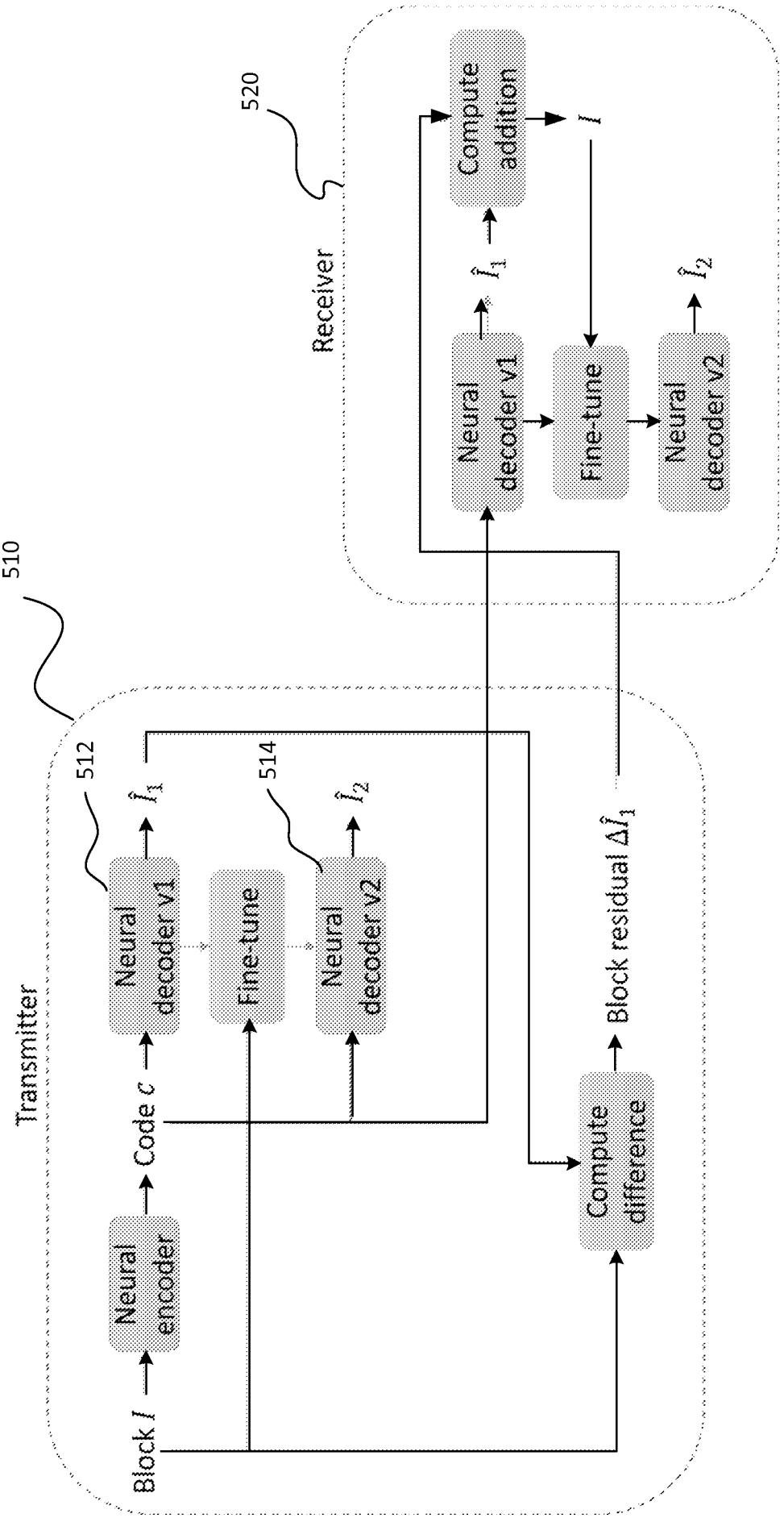
510
Transmitter
Block I
Neural encoder
Code c
512
Neural decoder v1
$\hat{I}_1$
Fine-tune
514
Neural decoder v2
$\hat{I}_2$
Compute difference
Block residual $\Delta\hat{I}_1$
520
Receiver
Neural decoder v1
$\hat{I}_1$
Compute addition
I
Fine-tune
Neural decoder v2
$\hat{I}_2$

Fig. 5

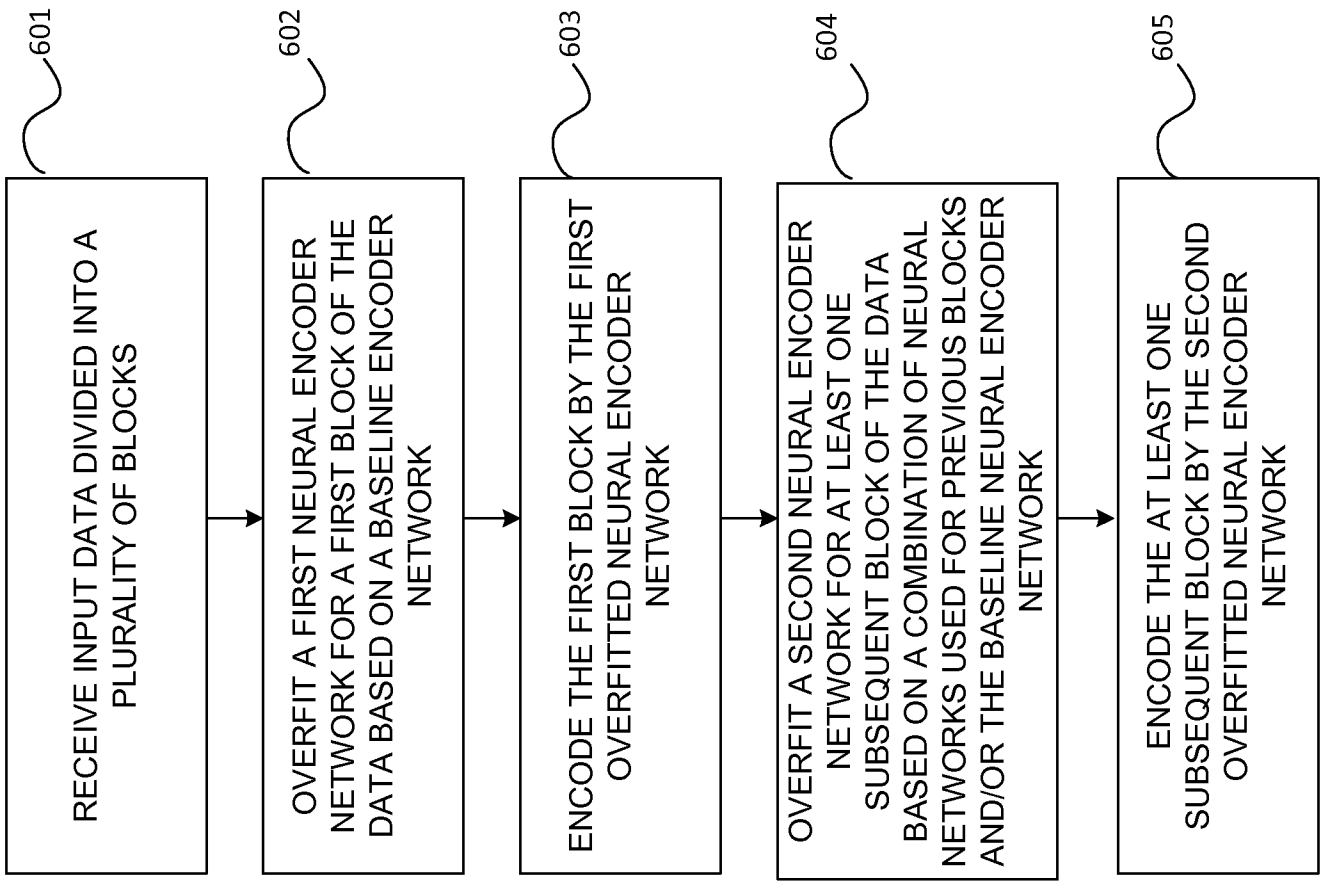
601
RECEIVE INPUT DATA DIVIDED INTO A PLURALITY OF BLOCKS
602
OVERFIT A FIRST NEURAL ENCODER NETWORK FOR A FIRST BLOCK OF THE DATA BASED ON A BASELINE ENCODER NETWORK
603
ENCODE THE FIRST BLOCK BY THE FIRST OVERFITTED NEURAL ENCODER NETWORK
604
OVERFIT A SECOND NEURAL ENCODER NETWORK FOR AT LEAST ONE SUBSEQUENT BLOCK OF THE DATA BASED ON A COMBINATION OF NEURAL NETWORKS USED FOR PREVIOUS BLOCKS AND/OR THE BASELINE NEURAL ENCODER NETWORK
605
ENCODE THE AT LEAST ONE SUBSEQUENT BLOCK BY THE SECOND OVERFITTED NEURAL ENCODER NETWORK

Fig. 6

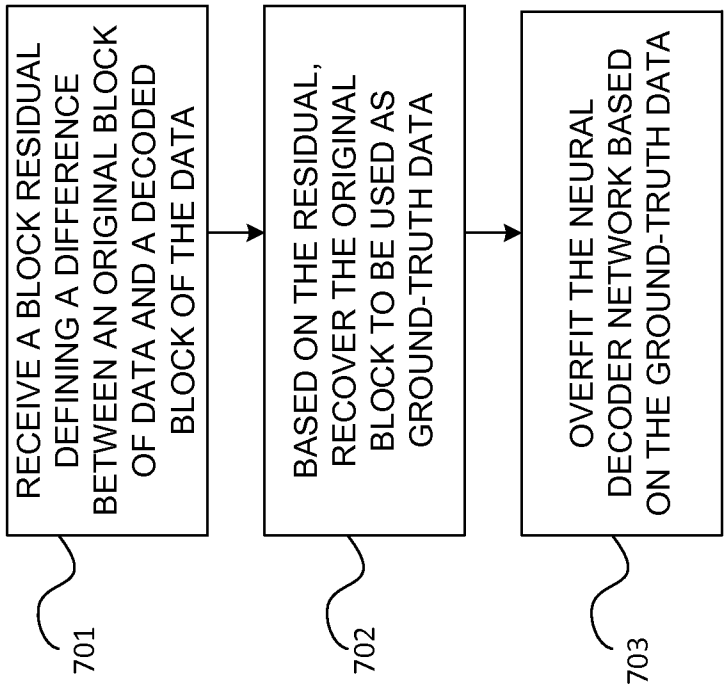
701
RECEIVE A BLOCK RESIDUAL DEFINING A DIFFERENCE BETWEEN AN ORIGINAL BLOCK OF DATA AND A DECODED BLOCK OF THE DATA
702
BASED ON THE RESIDUAL, RECOVER THE ORIGINAL BLOCK TO BE USED AS GROUND-TRUTH DATA
703
OVERFIT THE NEURAL DECODER NETWORK BASED ON THE GROUND-TRUTH DATA

Fig. 7

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant´s or agent´s file reference NC306555WO	**FOR FURTHER ACTION**	see Form PCT/ISA/220 as well as, where applicable, item 5 below.
International application No. PCT/FI2019/050483	International filing date *(day/month/year)* 20 June 2019 (20.06.2019)	(Earliest) Priority date *(day/month/year)* 02 July 2018 (02.07.2018)
Applicant NOKIA TECHNOLOGIES OY		

This international search report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This international search report consists of a total of 5 sheets.

☒ It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**
   a. With regard to the **language**, the international search was carried out on the basis of:
      ☒ the international application in the language in which it was filed.
      ☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).
   b. ☐ This international search report has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43.6*bis*(a)).
   c. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, see Box No. I.

2. ☐ **Certain claims were found unsearchable** (See Box No. II).

3. ☐ **Unity of invention is lacking** (See Box No. III).

4. With regard to the **title**,
   ☒ the text is approved as submitted by the applicant.
   ☐ the text has been established by this Authority to read as follows:

5. With regard to the abstract,
   ☐ the text is approved as submitted by the applicant.
   ☒ the text has been established, according to Rule 38.2, by this Authority as it appears in Box No. IV. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. With regard to the **drawings**,
   a. the figure of the **drawings** to be published with the abstract is Figure No. 3
      ☐ as suggested by the applicant.
      ☐ as selected by this Authority, because the applicant failed to suggest a figure.
      ☒ as selected by this Authority, because this figure better characterizes the invention.
   b. ☐ none of the figures is to be published with the abstract.

International application No.
PCT/FI2019/050483

Box No. IV **Text of the abstract (Continuation of item 5 of the first sheet)**

The embodiments relate to a method and to a technical equipment implementing the method for image, video, or audio encoding or decoding. The method comprises receiving input data divided into a plurality of blocks; overfitting a first neural encoder network for a first block of the data based on a baseline encoder network (310); encoding the first block by the first overfitted neural encoder network (305); overfitting a second neural encoder network for at least one subsequent block of the data based on a combination of neural encoder networks used for previous blocks (306) and/or the baseline encoder network; and encoding the at least one subsequent block by the second overfitted neural encoder network. Additionally, a block residual defining a difference between an original block of data and a decoded block of the data may be received on the decoder side.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2019/050483

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: H04N, G06T, G06N

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

FI, SE, NO, DK

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

EPODOC, EPO-Internal full-text databases, WPIAP, XPESP, XPI3E, Inspec, Internet, PRH Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018184123 A1 (TERADA KENGO [JP] et al.) 28 June 2018 (28.06.2018) abstract; paragraphs [0006], [0124], [0131], [0160]-[0164], [0170], [0196]-[0200], [0349], [0459]-[0460]; Figs. 1, 29, 31	1-16
X	AYTEKIN, C. et al. Block-optimized Variable Bit Rate Neural Image Compression. arXiv [online], 2018-05-28, [retrieved on 2019-09-27]. Retrieved from <http://arxiv.org/abs/1805.10887v1> abstract; section 2.2	1-11, 13-16
A	WO 2017036370 A1 (MEDIATEK INC [CN]) 09 March 2017 (09.03.2017) abstract; paragraphs [0030]-[0034]; Figs. 3-4	1-16
A	ZHANG, C. et al. Understanding Deep Learning Requires Rethinking Generalization. arXiv [online], 2017-02-26, [retrieved on 2019-06-13]. Retrieved from <https://arxiv.org/abs/1611.03530v2> see the whole document, especially abstract	1-16

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

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 September 2019 (30.09.2019)	01 October 2019 (01.10.2019)

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
Mika Inki
Telephone No. +358 29 509 5000

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2019/050483

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
US 2018184123 A1	28/06/2018	CN 107736027 A	23/02/2018
		EP 3310058 A1	18/04/2018
		WO 2016199330	05/04/2018
		KR 20180018544 A	21/02/2018
		WO 2016199330 A1	15/12/2016
WO 2017036370 A1	09/03/2017	CA 2997193 A1	09/03/2017
		CN 107925762 A	17/04/2018
		EP 3342164 A1	04/07/2018
		KR 20180052651 A	18/05/2018
		PH 12018500454 A1	10/09/2018
		US 2018249158 A1	30/08/2018

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2019/050483

CLASSIFICATION OF SUBJECT MATTER

IPC
G06T 9/00 (2006.01)
H04N 19/192 (2014.01)
H04N 19/196 (2014.01)
G06N 3/04 (2006.01)
H04N 19/176 (2014.01)
G06N 20/00 (2019.01)