(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 3 771 999 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
18.01.2023 Bulletin 2023/03

(21) Application number: **20161994.7**

(22) Date of filing: **10.03.2020**

(51) International Patent Classification (IPC):
G06F 17/15 *(2006.01)* ***G06N 3/04*** *(2006.01)*
G06N 3/063 *(2006.01)*

(52) Cooperative Patent Classification (CPC):
G06F 17/153; G06N 3/0454; G06N 3/063

(54) **METHOD AND APPARATUS FOR EXTRACTING IMAGE DATA IN PARALLEL FROM MULTIPLE CONVOLUTION WINDOWS, DEVICE, AND COMPUTER-READABLE STORAGE MEDIUM**

VERFAHREN UND VORRICHTUNG ZUR PARALLELEN EXTRAKTION VON BILDDATEN AUS MEHREREN FALTUNGSFENSTERN, VORRICHTUNG UND COMPUTERLESBARES SPEICHERMEDIUM

PROCÉDÉ ET APPAREIL PERMETTANT D'EXTRAIRE DES DONNÉES D'IMAGE EN PARALLÈLE À PARTIR DE PLUSIEURS FENÊTRES DE CONVOLUTION, DISPOSITIF ET SUPPORT D'ENREGISTREMENT LISIBLE PAR ORDINATEUR

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(30) Priority: **30.07.2019 CN 201910694475**

(43) Date of publication of application:
03.02.2021 Bulletin 2021/05

(73) Proprietor: **KunlunXin Technology (Beijing) Company Limited**
Haidian District
Beijing 100101 (CN)

(72) Inventors:
- **LIANG, Zihao**
  Beijing, 100085 (CN)
- **OUYANG, Jian**
  Beijing, 100085 (CN)

(74) Representative: **Lucke, Andreas et al**
Boehmert & Boehmert
Anwaltspartnerschaft mbB
Pettenkoferstrasse 22
80336 München (DE)

(56) References cited:
EP-A1- 3 480 740
WO-A1-2018/196863
WO-A1-2018/196863
WO-A1-2019/109795
WO-A1-2019/109795

- **Jzerman J. G.: "Customized low power processor for object recognition a programmable high performance low power TTA-SIMD accelerator for CNN-based object recognition", Master's thesis, 31 December 2016 (2016-12-31), XP055851911, Retrieved from the Internet: URL:https://pure.tue.nl/ws/portalfiles/portal/46944848/855329-1.pdf [retrieved on 2021-10-15]**
- **JOS IJZERMAN ET AL: "AivoTTA", EMBEDDED COMPUTER SYSTEMS, ACM, 2 PENN PLAZA, SUITE 701NEW YORKNY10121-0701USA, 15 July 2018 (2018-07-15), pages 28-37, XP058423983, DOI: 10.1145/3229631.3229637 ISBN: 978-1-4503-6494-2**

EP 3 771 999 B1

Description

TECHNICAL FIELD

[0001] Embodiments of the present disclosure generally relate to the field of image data processing technology, and more specifically to a method for extracting image data in parallel from multiple convolution windows by an accelerator device including multiple data processing units for transforming an image convolution operation into a multiplication of two-dimensional matrixes, a corresponding accelerator device, and a computer-readable storage medium.

BACKGROUND

[0002] Machine learning enables a machine to learn laws from a large amount of data like humans, thus generating a machine learning model that can complete some specific tasks. Artificial neural networks are a typical machine learning technology. An artificial neural network is created based on a human brain model, and allows a computer to learn through mass data by using various machine learning algorithms. Common artificial neural networks include convolutional neural networks (CNNs), recurrent neural networks (RNNs), and the like. Deep learning is also a type of machine learning, but the deep learning uses a deep neural network (DNN), so that the processing of a model is more complex, and the model understands data more deeply.

[0003] CNN is a feed-forward neural network containing convolutional calculation and having a deep structure, and is widely applied in the field of computer vision, especially image processing. From the perspective of a computer, an image is actually a two-dimensional or three-dimensional matrix. The CNN is used to extract features from a two-dimensional or three-dimensional array by convolution, pooling or the like, and identify the image. The CNN usually consists of an input layer, a convolutional layer, an activation function, a pooling layer, and a fully connected layer.

[0004] With the diversification of neural network models and the increase in computing power requirements, the industry has begun to develop deep learning accelerators in view of the factors such as performance and cost of conventional deep learning hardware platforms (such as a general-purpose processor and a graphics processing unit (GPU)). One of the hardware cores of the deep learning accelerator is matrix operation, and the operation of a matrix operation module depends on the upper level of data supply. In order to make full use of the computing power of the matrix operation module, efficient and flexible data supply is the focus of hardware design.

[0005] WO 2019/109795 A1 describes a method for dividing convolution input data and corresponding convolution kernels, wherein there are N convolution kernels and each of the N convolution kernels is sectioned into Y sectioned convolution kernels. The convolution input data is sectioned into Y pieces of sectioned convolution input data.

[0006] Methods based on vector-processors for CNN-based object recognition is known from Ijzerman et al: "AivoTTA", Embedded Computer Systems, ACM, 2 Penn Plaza, Suite 701, New York, USA, 15 July 2018 (2018-07-15), pages 28-37, XP058423983, DOI: 10.1145/13229613132296137, ISBN: 978-1-4503-6494-2 and from IJzerman J. G.: "Customized low power processor for object recognition a programmable high performance low power TTA-SIMD accelerator for CNN-based object recognition", Master's thesis, 31 December 2016 (2016-12-31), XP05585191.

SUMMARY

[0007] The present invention provides a method for extracting image data in parallel from multiple convolution windows by an accelerator device including multiple data processing units for transforming an image convolution operation into a multiplication of two-dimensional matrixes, a corresponding accelerator device, and a computer-readable storage medium.

[0008] A first aspect of the invention refers to a method for extracting image data in parallel from multiple convolution windows according to claim 1.

[0009] A second aspect of the invention refers to an apparatus for extracting image data in parallel from multiple convolution windows according to claim 7.

[0010] An electronic device related to the present invention but not directly corresponding to the subject-matter of the claims may include: one or more processors; and a storage apparatus for storing one or more programs, where the one or more programs, when executed by the one or more processors, cause the electronic device to implement the various methods and or processes according to embodiments of the invention.

[0011] A third aspect of the invention refers to a computer-readable storage medium according to claim 13.

[0012] Preferred embodiments of the invention are defined in the dependent claims. Other features of the present disclosure will become readily comprehensible through the following description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The above and other features, advantages and aspects of various embodiments of the present disclosure will become more apparent with reference to the accompanying drawings and detailed descriptions below. The same or similar reference signs in the drawings denote the same or similar elements.

Fig. 1 shows a schematic diagram of a convolution process in a convolutional neural network;
Fig. 2 shows a flowchart of a method for extracting image data concurrently from multiple convolution windows according to an embodiment of the invention;
Fig. 3 shows a schematic diagram of a process of extracting image data concurrently from multiple convolution windows according to an embodiment of the invention;
Fig. 4 shows a schematic diagram of an example architecture of an accelerator device for processing data concurrently according to an embodiment of the invention;
Fig. 5 shows a schematic diagram of an example process of extracting convolution data according to an embodiment of the invention;
Fig. 6 shows a schematic diagram of an example process for concurrent matrix transposition according to an embodiment of the invention;
Fig. 7 shows a block diagram of an apparatus for extracting image data concurrently from multiple convolution windows according to an embodiment of the invention; and
Fig. 8 shows a block diagram of an electronic device capable of implementing multiple embodiments of the invention.

DETAILED DESCRIPTION OF EMBODIMENTS

[0014] The embodiments of the invention will be described in more detail below with reference to the accompanying drawings. It should be understood that the drawings and embodiments of the invention are merely illustrative, but are not intended to limit the scope of the invention.

[0015] Conventionally, in the process of image convolution processing, a convolution kernel is slid on an image, and pixels of a convolution window are extracted each time and output. However, the conventional method is to extract image data from different convolution windows serially, so data conversion cannot be performed efficiently, which affects the processing performance. In addition, the conventional scheme also performs matrix transposition serially. Therefore, the shortcomings of the related art mainly lie in that the concurrence of hardware cannot be fully exerted while the flexibility is ensured, only one number or a group of numbers is operated each time, and data conversion cannot be performed efficiently, thus limiting the performance of subsequent calculation.

[0016] Therefore, the embodiments of the invention propose a solution for extracting image data in parallel from multiple convolution windows by an accelerator device including multiple data processing units for transforming an image convolution operation into a multiplication of two-dimensional matrixes. According to the invention, during the extraction of convolution data, multiple data processing units are used to extract image data from multiple convolution windows in parallel, which improves the speed of data extraction, thereby improving the processing efficiency of image convolution. In addition, some embodiments of the invention also propose a solution of concurrent convolution kernel transposition, where multiple columns in the convolutions kernel matrix are extracted in parallel by multiple data processing units, which improves the speed of matrix transposition. Some example implementations of the embodiments of the invention will be described in detail below with reference to Figs. 1-8.

[0017] Fig. 1 shows a schematic diagram of a convolution process 100 in a convolutional neural network. The convolutional neural network discovers some features of an image by image convolution, for example, searches edges of an object in the image, enhances or weakens an effect of the image, such as blurring, sharpening or embossing effect of the image.

[0018] Fig. 1 illustrates an example process of convolving an image 110 by a convolution kernel 120, where the convolution kernel 120 may be a 3×3 two-dimensional matrix. It should be understood that multiple convolution kernels may be used to convolve the image. The idea of image convolution is to weight the values of single pixels in the input image (for example, the image 110) by the values of surrounding adjacent pixels, and the new pixel values generated by the weighting operation can generate a new output image (for example, an image 130).

[0019] The convolution kernel 120 obtains convolution data by sliding each convolution window in the image 110. As shown in Fig. 1, first, the convolution kernel is slid to the first convolution window 111 in the image 110, the products of pixels in the convolution window 111 and the convolution kernel 120 are accumulated (as shown by 121) to generate a convolution output 131, and the convolution output 131 is stored in the image 130. For example, elements are multiplied and then added, and the obtained value is placed at the position of the first element of the output image matrix.

[0020] After the convolution of the convolution window 111 is completed, the convolution kernel is slid to the right by 1 distance or more distances. This distance is called a stride, which may be preset. Next, as shown by arrow 140 in Fig.

1, for the second convolution window 112 in the image 110, the products of pixels in the convolution window 112 and the convolution kernel 120 are accumulated (as shown by 122) to generate a convolution output 132, and the convolution output 132 is stored in the image 130. Then, the convolution process is repeated until the convolution kernel 120 is slid throughout all the convolution windows in the image 110, thus generating a convolved image 130. However, the data is serially extracted and sequentially calculated in the convolution process described in Fig. 1, so that the convolution process is slow.

[0021] Fig. 2 shows a flowchart of a method 200 for extracting image data in parallel from multiple convolution windows according to an embodiment of the invention. It should be understood that the method 200 may be performed by a dedicated accelerator device (such as an artificial intelligence (AI) chip), a general-purpose computer, or other dedicated computing devices.

[0022] In block 202, an image is divided into multiple groups of convolution windows, where the multiple groups of convolution windows include a first group of convolution windows and a second group of convolution windows. According to the invention, the image is divided into multiple groups of convolution windows (each group of convolution windows includes P convolution windows) according to the number (e.g., P) of available data processing units, so that each group of convolution windows can be processed in parallel by multiple data processing units.

[0023] In block 204, image data is extracted in parallel from multiple convolution windows in the first group of convolution windows by using multiple data processing units. For example, the first group of convolution windows may include P convolution windows, and image data is extracted in parallel from the P convolution windows by using P data processing units in an acceleration device (such as an AI chip), that is. According to the invention, each processing unit extracts image data from a corresponding convolution window. In this way, the extraction speed of the image data in the convolution windows is improved.

[0024] In block 206, after the extraction of image data from the first group of convolution windows is completed, image data is extracted in parallel from multiple convolution windows in the second group of convolution windows by using the multiple data processing units. Generally, the number of convolution windows in an image may be much greater than the number of data processing units, so the data needs to be extracted in parallel in sections. For example, after extracting the image data in parallel from the P convolution windows, the P data processing units extract data from next P convolution windows. According to the invention, this step is repeated till the image data in all the convolution windows of the image is extracted.

[0025] Therefore, according to the embodiment of the invention, during the extraction of convolution data, multiple data processing units are used to extract image data in parallel from multiple convolution windows, which improves the speed of data extraction, thereby improving the processing efficiency of image convolution.

[0026] Fig. 3 shows a schematic diagram of a process 300 of extracting image data in parallel from multiple convolution windows according to an embodiment of the invention. As shown in Fig. 3, convolution windows 311, 312, 313 in an image 310 may be processed in parallel by data processing units 321, 322, 323, respectively, and corresponding data 331, 332, 333 (which according to the invention are one-dimensional vectors, respectively) in the convolution windows 311, 312, 313 may be extracted concurrently. It should be understood that, for the purpose of clear illustration, the stride of the convolution windows in Fig. 3 is 3, so that the three convolution windows 311, 312, 313 do not repeat. However, the stride may also be set to 1 or other values, so that different convolution windows may have repeated pixels. In addition, for simplicity, the image 310 of only one color channel is shown in Fig. 3. However, the image 310 may alternatively include multiple color channels.

[0027] Fig. 4 shows a schematic diagram of an example architecture 400 of an accelerator device for processing data concurrently according to an embodiment of the invention. As shown in Fig. 4, the example architecture 400 may include a processor 410, a source memory 420, a target memory 425, a data conversion module 431, a scheduler 470, and the like. The data conversion module 431 may serve as a co-processor, and includes an instruction storage unit 430, an instruction decoding unit 440, a control unit 450, a synchronization unit 460, a data reading unit 480, and multiple data processing units 490, where the multiple data processing units 490 may include, for example, P data processing units 491, 492, 493, 494, and 499.

[0028] The source memory 420 and the target memory 425 are respectively an input memory and an output memory, and may be off-chip memories (such as double data rate synchronous dynamic random access memories (DDRs)) or on-chip memories (such as static random access memories (SRAMs)), where the source memory 420 and the target memory 425 may be different memories or the same memory.

[0029] The instruction storage unit 430 is used to store an instruction received from the processor 410 for data conversion. The type of the instruction may include, but is not limited to, a parameter configuration instruction, a transposition instruction, a convolution data extraction instruction, a synchronization instruction, or the like. The parameter configuration instruction is used to configure parameters. The parameters include, but are not limited to: data type, scale of a transposed matrix, scale of an convolved image, scale of a convolution kernel, convolution stride, number of edge filling pixels (pads), etc. The transposition instruction is used to configure an initial address of the source memory 420, an initial address of the target memory 425, a length of transposed data, etc. The convolution data extraction instruction is used

to configure an initial address of the source memory 420, an initial address of the target memory 425, a length of extracted data, etc. The synchronization instruction is used to ensure that all the instructions before the instruction are executed and the data is stored in disks, so that the scheduler 470 synchronizes respective modules.

[0030] The instruction decoding unit 440 is used to read, when it is detected that the instruction storage unit 430 is not empty and has a currently executable instruction, the instruction from the instruction storage unit 430, parse the instruction, and send the parsed content to the control unit 450. The control unit 450 generates a corresponding control signal according to the configured parameters, and the control content includes, but is not limited to, a read request behavior of the data reading unit 480, behaviors of the data processing units 490, and a behavior of the synchronization unit 460.

[0031] The data reading unit 480 sends a read request to the source memory 420 according to the control signal of the control unit 450, and transmits the read data to the multiple data processing units 490. The multiple data processing units 490 extract a specific portion of the data from the data reading unit 480 according to the control signal of the control unit 450, and write the data to the target memory 425. According to the embodiment of the invention, the multiple data processing units 490 may extract image data in parallel from multiple convolution windows, and may also transpose multiple columns in a matrix in parallel, thereby improving the speed of data conversion.

[0032] The synchronization unit 460 outputs a synchronization completion signal to the external scheduler 470 after receiving a synchronization request and detecting that the current instruction is completed and the data is stored in disks. It should be understood that the example architecture 400 of the accelerator device is only an example architecture including multiple data processing units 490, and other acceleration device having multiple data processing units may also be used with the embodiments of the invention.

[0033] Fig. 5 shows a schematic diagram of an example process 500 of extracting convolution data according to an embodiment of the invention. As shown in Fig. 5, the image 510 has a width of W, a height of H, and a channel depth of C, and each convolution window has a width of S and a height of R (the size of the convolution windows is 3×3 in the example of Fig. 5). The accelerator device for image convolution includes multiple data processing units 520, for example, includes P data processing units 521, 522, 523, and 529. According to an embodiment of the invention, the multiple data processing units may extract image data in parallel from multiple convolution windows.

[0034] Referring to Fig. 5, the data processing unit 521 is used to extract image data from the convolution window 511. The data processing unit 521 first extracts a first row of data in a first channel (the respective data processing units extract the first row of data in the corresponding convolution windows in parallel), then a second row of data in the first channel, and a third row of data in the first channel. So far, the extraction of data from the first channel in the convolution window 511 in the example of Fig. 5 is completed. Next, the data processing unit 521 extracts all image data from a second channel of the convolution window 511, all image data from a third channel of the convolution window 511, and all image data from a fourth channel of the convolution window 511, thereby completing the data extraction process for the convolution window 511. As shown in Fig. 5, the extracted data 530 includes data 531 of the first channel (including three rows, in a total of 9 values of the first channel), data of the second channel, data of the third channel, and data of the fourth channel 534. According to the invention, since the P data processing units extract data in parallel, the P data processing units can complete the extraction of all image data in parallel from the first P convolution windows .

[0035] Next, the multiple data reading units 520 read the data of the subsequent P windows in parallel by the same method as above. Finally, the extraction of data corresponding to all the convolution windows in the image 510 is completed. Since the P data processing units extract the convolution data in parallel, each data processing unit needs to acquire data of the corresponding convolution window according to the stride parameter, and this part of control behavior can be completed by the control unit.

[0036] According to the invention, since the extracted data of one convolution window is continuously stored in the target memory, the image data in a three-dimensional convolution window having a scale of CxRxS is represented by a one-dimensional vector having a length of CxRxS on the target memory after being extracted by the data processing unit. Assuming the data of N convolution windows is extracted from the image 510, a two-dimensional matrix having N rows and CxRxS columns is finally stored on the target memory. The convolution kernel is represented by a two-dimensional matrix having F rows and CxRxS columns. If the convolution kernel is transposed into a two-dimensional matrix having CxRxS rows and F columns, the complex image convolution operation is transformed into a multiplication of two two-dimensional matrixes. As shown in the following formula (1), D represents an image data matrix, and W represents a weight data matrix. The image data contained in a convolution window is, for example, the left dotted box (i.e., a one-dimensional vector having a length of CxRxS), and the weight data contained in a convolution kernel is, for example, the right dotted box. In this way, the matrix operation efficiency in the convolution operation can be further improved.

$$\begin{pmatrix} D_{0,0} & D_{0,1} & \cdots & D_{0,C*R*S-1} \\ D_{1,0} & D_{1,1} & \cdots & D_{1,C*R*S-1} \\ \cdots & \cdots & \cdots & \cdots \\ D_{N-1,0} & D_{N-1,1} & \cdots & D_{N-1,C*R*S-1} \end{pmatrix} \cdot \begin{pmatrix} W_{0,0} & W_{0,1} & \cdots & W_{0,F-1} \\ W_{1,0} & W_{1,1} & \cdots & W_{1,F-1} \\ \cdots & \cdots & \cdots & \cdots \\ W_{C*R*S-1,0} & W_{C*R*S-1,1} & \cdots & W_{C*R*S-1,F-1} \end{pmatrix} \quad (1)$$

[0037] Fig. 6 shows a schematic diagram of an example process 600 for concurrent matrix transposition according to an embodiment of the invention. As shown in Fig. 6, it is assumed that an MxN matrix 610 needs to be transposed. Referring to the data conversion module including P concurrent data processing units as described in Fig. 4, the matrix 610 is divided into blocks by P columns as a granularity, that is, the first block includes first P columns, the second block includes second P columns, and so on.

[0038] As shown in Fig. 6, multiple data processing units 620 include P data processing units, such as data processing units 621, 622, 623, 629, and the like. Each time the data reading unit reads a row of data of the matrix, each data processing unit process a corresponding column in the row of data in parallel, for example, the data processing unit 621 processes data in the first column (column 0), the data processing unit 622 processes data in the second column (column 1), the data processing unit 623 processes data in the third column (column 2), and the data processing unit 629 processes data in the P column (column P-1).

[0039] After processing the P columns of the first block in parallel, the multiple data processing units 620 continue to process P columns of data in next block until the entire matrix 621 is transposed to generate a transposed matrix 630. As shown in Fig. 6, the data processing unit 621 transposes the first column in the matrix 610 into the first row in the matrix 630, the data processing unit 622 transposes the second column in the matrix 610 into the second row in the matrix 630, and the data processing unit 629 transposes the P-th column in the matrix 610 into the P-th row in the matrix 630. In some implementations, the control unit needs to maintain the write addresses of respective target memories of the P data processing units according to the parameters of instruction configuration and the initial addresses of the target memories.

[0040] Therefore, according to the invention, during the extraction of convolution data, multiple data processing units are used to extract image data in parallel from multiple convolution windows, which can improve the speed of data extraction, thereby improving the processing efficiency of image convolution. In addition, the multiple data processing units extract columns in a matrix in parallel in some embodiments of the invention, which can improve the speed of matrix transposition.

[0041] Fig. 7 shows a block diagram of an apparatus 700 for extracting image data in parallel from multiple convolution windows according to an implementation of the invention. As shown in Fig. 7, the apparatus 700 includes a convolution window group division module 710, a first concurrent extraction module 720, and a second concurrent extraction module 730. The convolution window group division module 710 is configured to divide an image into multiple groups of convolution windows, where the multiple groups of convolution windows include a first group of convolution windows and a second group of convolution windows. The first concurrent extraction module 720 is configured to extract image data in parallel from multiple convolution windows in the first group of convolution windows by using multiple data processing units. The second concurrent extraction module 730 is configured to extract, in response to completing the extraction of image data from the first group of convolution windows, image data in parallel from multiple convolution windows in the second group of convolution windows by using the multiple data processing units.

[0042] In some embodiments, the first group of convolution windows includes a first convolution window and a second convolution window, and the first concurrent extraction module 720 includes: a first data extraction module, configured to extract image data from the first convolution window by using a first data processing unit; and a second data extraction module, configured to extract image data from the second convolution window by using a second data processing unit.

[0043] In some embodiments, the first data extraction module includes: a first extraction module, configured to extract a first row of image data from a first channel in the first convolution window; a second extraction module, configured to extract a second row of image data from the first channel in the first convolution window; and a third extraction module, configured to extract a third row of image data from the first channel in the first convolution window.

[0044] In some embodiments, the first data extraction module further includes: a second channel extraction module configured to, in response to completing the extraction of all image data from the first channel in the first convolution window: extract a first row of image data from a second channel in the first convolution window; extract a second row of image data from the second channel in the first convolution window; and extract a third row of image data from the second channel in the first convolution window.

[0045] In some embodiments, the first data extraction module further includes: a data representation module, configured to represent, in response to completing the extraction of all image data from all channels in the first convolution window,

all the image data in the first convolution window by using a one-dimensional vector, where the length of the one-dimensional vector is the product of the number of channels in the image, the number of rows in each convolution window, and the number of columns in each convolution window.

[0046] According to the invention, the apparatus 700 further includes: a data storage module, configured to store all image data in the multiple groups of convolution windows into a target memory by using a two-dimensional matrix, where the number of rows in the two-dimensional matrix is the number of all convolution windows in the multiple groups of convolution windows, and the number of columns in the two-dimensional matrix is the product of the number of channels in the image, the number of rows in each convolution window, and the number of columns in each convolution window.

[0047] In some embodiments, the apparatus 700 further includes: a block division module, configured to divide a matrix into multiple blocks in columns, the multiple blocks including a first block and a second block; a first concurrent transposition module, configured to transpose multiple columns of data in the first block in parallel by using the multiple data processing units; and a second concurrent transposition module, configured to transpose, in response to completing the transposition of multiple columns of data in the first block, multiple columns of data in the second block in parallel by using the multiple data processing units.

[0048] In some embodiments, the first concurrent transposition module includes: a first matrix transposition module, configured to transpose a first column of data in the first block by using the first data processing unit in the multiple data processing units; and a second matrix transposition module, configured to transpose a second column of data in the second block by using the second data processing unit in the multiple data processing units.

[0049] In some embodiments, the block division module includes: a second block division module, configured to divide the matrix into the multiple blocks based on the number of the multiple data processing units.

[0050] It should be understood that the convolution window group division module 710, the first concurrent extraction module 720, and the second concurrent extraction module 730 shown in Fig. 7 may be included in a single or multiple electronic devices. Moreover, it should be understood that the modules illustrated in Fig. 7 may perform the steps and/or operations in the methods and/or processes according to the embodiments of the invention.

[0051] Therefore, the embodiments of the invention allow to flexibly support matrix transposition of various scales and convolution window extraction of images, and can efficiently provide data by fully using the characteristic of concurrence of hardware so as to exert the performance of a matrix operation module. The embodiments of the invention ensure the flexibility of data conversion through programmability, and efficiently convert data by means of concurrent operation of multiple processing units. In addition, the embodiments of the invention can reuse the same set of hardware structure for transposition and convolution, thereby reducing the hardware overhead of final implementation.

[0052] Therefore, the benefits of some embodiments of the invention may include, but are not limited to: multiple data processing units operate in parallel to efficiently complete data conversion; a processor transmits a parameter configuration instruction to flexibly configure parameters, which can adapt to multiple scales of data conversion; the complex convolution operation can be transformed into a simple matrix multiplication by the data conversion method of convolution data extraction; and the transposition and extraction of convolution data can be completed by the same set of hardware structure, which saves hardware resources.

[0053] Fig. 8 shows a schematic block diagram of an exemplary device 800 that can be used to implement the embodiments of the invention. It should be understood that the device 800 may be used to implement the apparatus 700 for extracting image data in parallel from multiple convolution windows according to the embodiments of the invention. As shown in the figure, the device 800 includes a central processing unit (CPU) 801, which may execute various appropriate operations and processing based on computer program instructions stored in a read-only memory (ROM) 802 or computer program instructions loaded from a storage unit 808 to a random access memory (RAM) 803. The RAM 803 may also store various programs and data required by the operations of the device 800. The CPU 801, the ROM 802, and the RAM 803 are connected to each other through a bus 804. As shown in Fig. 8, an input/output (I/O) interface 805 is also connected to the bus 804.

[0054] A plurality of components in the device 800 are coupled to the I/O interface 805, including: an input unit 806, such as a keyboard or a mouse; an output unit 807, such as various types of displays, or speakers; the storage unit 808, such as a disk or an optical disk; and a communication unit 809 such as a network card, a modem, or a wireless communication transceiver. The communication unit 809 allows the device 800 to exchange information/data with other devices over a computer network such as the Internet and/or various telecommunication networks.

[0055] The processing unit 801 performs the various methods and processes described above, such as the method 200. For example, the method may be implemented as a computer software program that is tangibly embodied in a machine readable medium, such as the storage unit 808. In some implementations, some or all of the computer programs may be loaded and/or installed onto the device 800 via the ROM 802 and/or the communication unit 809. When a computer program is loaded into the RAM 803 and executed by the CPU 801, one or more of the actions or steps of the method described above may be performed. Alternatively, the CPU 801 may be configured to perform the method by any other suitable means (e.g., by means of firmware).

[0056] The functions described herein above may be performed, at least in part, by one or more hardware logic

components. For example, and without limitation, exemplary types of hardware logic components that may be used include: Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), Application Specific Standard Product (ASSP), System on Chip (SOC), Complex Programmable Logic Device (CPLD), and the like.

[0057] Program codes for implementing the method of the invention may be written in any combination of one or more programming languages. These program codes may be provided to a processor or controller of a general purpose computer, special purpose computer or other programmable data processing apparatus such that the program codes, when executed by the processor or controller, enables the functions/operations specified in the flowcharts and/or block diagrams being implemented. The program codes may execute entirely on the machine, partly on the machine, as a stand-alone software package partly on the machine and partly on the remote machine, or entirely on the remote machine or server.

[0058] In the context of the invention, the machine readable medium may be a tangible medium that may contain or store programs for use by or in connection with an instruction execution system, apparatus, or device. The machine readable medium may be a machine readable signal medium or a machine readable storage medium. The machine readable medium may include, but is not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples of the machine readable storage medium may include an electrical connection based on one or more wires, portable computer disk, hard disk, random access memory (RAM), read only memory (ROM), erasable programmable read only memory (EPROM or flash memory), optical fiber, portable compact disk read only memory (CD-ROM), optical storage device, magnetic storage device, or any suitable combination of the foregoing.

Claims

1. A method for extracting image data in parallel from multiple convolution windows by an accelerator device including multiple data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) for transforming an image convolution operation into a multiplication of two-dimensional matrixes, the method comprising:

   dividing (202) an image (310, 510) into N convolution windows grouped into multiple groups of convolution windows, wherein each group of convolution windows includes a number (P) of convolution windows corresponding to the number of available data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529);
   extracting (204) image data in parallel from the (P) convolution windows (311, 312, 313, 511) in a first group of convolution windows of the multiple groups of convolution windows by using the available data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529), wherein extracting image data from a convolution window (511) by a corresponding processing unit (521, 522, 523, 529) comprises representing corresponding image data by using a one-dimensional vector having a length of Cx Rx S, wherein C is a number of channels in the image (310, 510), R is a number of rows in each convolution window (511), and S is a number of columns in each convolution window (511);
   extracting (206), in response to completing the extraction of image data from the first group of convolution windows, image data in parallel from the (P) convolution windows in a second group of convolution windows of the multiple groups of convolution windows by using the available data processing units (321, 322, 323, 491, 493, 521, 522, 523, 529),
   wherein the extracting (206) the image data is repeated until the image data in all N convolution windows of the image (310, 510) is extracted; and
   storing all extracted image data in the multiple groups of convolution windows into a target memory by using a two-dimensional matrix having N rows and CxRxS columns, wherein a convolution kernel represented by a two-dimensional matrix having F rows and CxRxS columns is transposed into a two-dimensional matrix having CxRxS rows and F columns, so that an image convolution operation is transformed into a multiplication of the two two-dimensional matrixes.

2. The method according to claim 1, wherein the available data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) comprise a first data processing unit and a second data processing unit, the first group of convolution windows comprises a first convolution window and a second convolution window, and the extracting (204) image data in parallel from the (P) convolution windows (311, 312, 313, 511) in the first group of convolution windows by using the multiple data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) comprises:

   extracting image data from the first convolution window by using the first data processing unit; and
   extracting image data from the second convolution window by using the second data processing unit.

3. The method according to claim 2, wherein the extracting image data from the first convolution window by using the first data processing unit comprises:

extracting a first row of image data from a first channel in the first convolution window;
extracting a second row of image data from the first channel in the first convolution window; and
extracting a third row of image data from the first channel in the first convolution window;
preferably, the extracting image data from the first convolution window by using the first data processing unit further comprises:
in response to completing the extraction of all image data from the first channel in the first convolution window:

extracting a first row of image data from a second channel in the first convolution window;
extracting a second row of image data from the second channel in the first convolution window; and
extracting a third row of image data from the second channel in the first convolution window.

4. The method according to claim 3, wherein the extracting image data from the first convolution window by using the first data processing unit further comprises:
representing, in response to completing the extraction of all image data from all channels in the first convolution window, all the image data in the first convolution window by using a one-dimensional vector, a length of the one-dimensional vector being a product of the number C of channels in the image (310, 510), the number R of rows in each convolution window (511), and the number S of columns in each convolution window (511).

5. The method according to any one of claims 1-4, further comprising:

dividing the convolution kernel into multiple blocks in columns, the multiple blocks comprising a first block and a second block;
transposing multiple columns of data in the first block in parallel by using the available data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529); and
transposing, in response to completing the transposition of multiple columns of data in the first block, multiple columns of data in the second block in parallel by using the available data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529); preferably, the transposing multiple columns of data in the first block in parallel by using the multiple data processing units comprises (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529):

transposing a first column of data in the first block by using a first data processing unit in the available data processing units; and
transposing a second column of data in the second block by using a second data processing unit in the available data processing units.

6. The method according to claim 5, wherein the dividing of the convolution kernel into multiple blocks in columns comprises:

dividing the convolution kernel into the multiple blocks based on a number of the available data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529).

7. An accelerator device for image convolution including multiple data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529), configured to extract image data in parallel from multiple convolution windows for transforming image convolution operation into a multiplication of two-dimensional matrixes, the accelerator device further comprising:

a convolution window group division module (710), configured to divide an image (310, 510) into N convolution windows grouped into multiple groups of convolution windows, wherein each group of convolution windows includes a number (P) of convolution windows corresponding to the number of available data processing units (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529);
a first concurrent extraction module (720), configured to extract image data in parallel from the (P) convolution windows in a first group of convolution windows of the multiple groups of convolution windows by using the available data processing units (321, 322, 323, 491, 493, 521, 522, 523, 529), wherein each processing unit (321, 322, 323, 491, 493, 521, 522, 523, 529) extracts image data from a corresponding convolution window including representing the corresponding image data by using a one-dimensional vector having a length of

CxRxS, wherein C is a number of channels in the image (310, 510), R is a number of rows in each convolution window (511), and S is a number of columns in each convolution window (511);
a second concurrent extraction module (730), configured to extract, in response to completing the extraction of image data from the first group of convolution windows, image data in parallel from the (P) convolution windows in a second group of convolution windows of the multiple groups of convolution windows by using the available data processing units (321, 322, 323, 491, 493, 521, 522, 523, 529), and to repeat the extracting the image data until the image data in all N convolution windows of the image (310, 510) is extracted; and
a data storage module, configured to store all image data in the multiple groups of convolution windows into a target memory by using a two-dimensional matrix having N rows and CxRxS columns, wherein N is a number of all convolution windows in the multiple groups of convolution windows,
wherein the accelerator device is further configured to transpose a convolution kernel represented by a two-dimensional matrix having F rows and CxRxS columns into a two-dimensional matrix having CxRxS rows and F columns, so that image convolution operation is transformed into a multiplication of the two two-dimensional matrixes.

8. The accelerator device according to claim 7, wherein the available data processing units (321, 322, 323, 491, 493, 521, 522, 523, 529) comprise a first data processing unit and a second data processing unit, the first group of convolution windows comprises a first convolution window and a second convolution window, and the first concurrent extraction module comprises:

a first data extraction module, configured to extract image data from the first convolution window by using the first data processing unit; and
a second data extraction module, configured to extract image data from the second convolution window by using the second data processing unit.

9. The accelerator device according to claim 8, wherein the first data extraction module comprises:

a first extraction module, configured to extract a first row of image data from a first channel in the first convolution window;
a second extraction module, configured to extract a second row of image data from the first channel in the first convolution window; and
a third extraction module, configured to extract a third row of image data from the first channel in the first convolution window;
preferably, the first data extraction module further comprises:
a second channel extraction module configured to, in response to completing the extraction of all image data from the first channel in the first convolution window:

extract a first row of image data from a second channel in the first convolution window;
extract a second row of image data from the second channel in the first convolution window; and
extract a third row of image data from the second channel in the first convolution window.

10. The accelerator device according to claim 9, wherein the first data extraction module further comprises:
a data representation module, configured to represent, in response to completing the extraction of all image data from all channels in the first convolution window, all the image data in the first convolution window by using a one-dimensional vector, a length of the one-dimensional vector being a product of the number C of channels in the image (310, 510), the number R of rows in each convolution window (511), and the number S of columns in each convolution window (511).

11. The accelerator device according to any one of claims 7-10, further comprising:

a block division module, configured to divide the convolution kernel into multiple blocks in columns, the multiple blocks comprising a first block and a second block;
a first concurrent transposition module, configured to transpose multiple columns of data in the first block in parallel by using the available data processing units (321, 322, 323, 491, 493, 521, 522, 523, 529); and
a second concurrent transposition module, configured to transpose, in response to completing the transposition of multiple columns of data in the first block, multiple columns of data in the second block in parallel by using the available data processing units (321, 322, 323, 491, 493, 521, 522, 523, 529);
preferably, the first concurrent transposition module comprises:

a first matrix transposition module, configured to transpose a first column of data in the first block by using the first data processing unit in the available data processing units (321, 322, 323, 491, 493, 521, 522, 523, 529); and
a second matrix transposition module, configured to transpose a second column of data in the second block by using the second data processing unit in the available data processing units (321, 322, 323, 491, 493, 521, 522, 523, 529).

12. The accelerator device according to claim 11, wherein the block division module comprises:
a second block division module, configured to divide the matrix into the multiple blocks based on a number of the available data processing units (321, 322, 323, 491, 493, 521, 522, 523, 529).

13. A computer-readable storage medium, storing a computer program thereon, wherein when the program is executed by a processor, the method according to any one of claims 1-6 is implemented.

Patentansprüche

1. Verfahren zum parallelen Extrahieren von Bilddaten aus mehreren Faltungsfenstern durch eine Beschleunigungsvorrichtung, die mehrere Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) zum Transformieren einer Bildfaltungsoperation in eine Multiplikation von zweidimensionalen Matrizen enthält, wobei das Verfahren Folgendes umfasst:

Aufteilen (202) eines Bildes (310, 510) in N Faltungsfenster, die in mehrere Gruppen von Faltungsfenstern gruppiert sind, wobei jede Gruppe von Faltungsfenstern eine Anzahl (P) von Faltungsfenstern enthält, die der Anzahl von verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) entspricht;
Extrahieren (204) von Bilddaten parallel aus den (P) Faltungsfenstern (311, 312, 313, 511) einer ersten Gruppe von Faltungsfenstern der mehreren Gruppen von Faltungsfenstern unter Verwendung der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529), wobei das Extrahieren von Bilddaten aus einem Faltungsfenster (511) durch eine entsprechende Verarbeitungseinheit (521, 522, 523, 529) das Darstellen entsprechender Bilddaten unter Verwendung eines eindimensionalen Vektors mit einer Länge CxRxS umfasst, wobei C eine Anzahl von Kanälen in dem Bild (310, 510) ist, R eine Anzahl von Zeilen in jedem Faltungsfenster (511) ist, und S eine Anzahl von Spalten in jedem Faltungsfenster (511) ist;
Extrahieren (206) von Bilddaten parallel aus den (P) Faltungsfenstern einer zweiten Gruppe von Faltungsfenstern der mehreren Gruppen von Faltungsfenstern unter Verwendung der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 521, 522, 523, 529) als Reaktion auf den Abschluss der Extraktion von Bilddaten aus der ersten Gruppe von Faltungsfenstern, wobei das Extrahieren (206) der Bilddaten wiederholt wird, bis die Bilddaten in allen N Faltungsfenstern des Bildes (310, 510) extrahiert sind; und
Speichern aller extrahierten Bilddaten in den mehreren Gruppen von Faltungsfenstern in einem Zielspeicher unter Verwendung einer zweidimensionalen Matrix mit N Zeilen und CxRxS Spalten, wobei ein durch eine zweidimensionale Matrix mit F Zeilen und CxRxS Spalten dargestellter Faltungskern in eine zweidimensionale Matrix mit CxRxS Zeilen und F Spalten transponiert wird, so dass eine Bildfaltungsoperation in eine Multiplikation der beiden zweidimensionalen Matrizen umgewandelt wird.

2. Verfahren nach Anspruch 1, wobei die verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) eine erste Datenverarbeitungseinheit und eine zweite Datenverarbeitungseinheit umfassen, die erste Gruppe von Faltungsfenstern ein erstes Faltungsfenster und ein zweites Faltungsfenster umfasst, und das parallele Extrahieren (204) von Bilddaten aus den (P) Faltungsfenstern (311, 312, 313, 511) der ersten Gruppe von Faltungsfenstern unter Verwendung der mehreren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) Folgendes umfasst:

Extrahieren von Bilddaten aus dem ersten Faltungsfenster unter Verwendung der ersten Datenverarbeitungseinheit; und
Extrahieren von Bilddaten aus dem zweiten Faltungsfenster unter Verwendung der zweiten Datenverarbeitungseinheit.

3. Verfahren nach Anspruch 2, wobei das Extrahieren von Bilddaten aus dem ersten Faltungsfenster unter Verwendung der ersten Datenverarbeitungseinheit Folgendes umfasst:

Extrahieren einer ersten Reihe von Bilddaten aus einem ersten Kanal in dem ersten Faltungsfenster;
Extrahieren einer zweiten Reihe von Bilddaten aus dem ersten Kanal in dem ersten Faltungsfenster; und
Extrahieren einer dritten Reihe von Bilddaten aus dem ersten Kanal in dem ersten Faltungsfenster;
wobei das Extrahieren von Bilddaten aus dem ersten Faltungsfenster unter Verwendung der ersten Datenverarbeitungseinheit vorzugsweise ferner Folgendes umfasst:
als Reaktion auf den Abschluss der Extraktion aller Bilddaten aus dem ersten Kanal in dem ersten Faltungsfenster:
Extrahieren einer ersten Reihe von Bilddaten aus einem zweiten Kanal in dem ersten Faltungsfenster;
Extrahieren einer zweiten Reihe von Bilddaten aus dem zweiten Kanal in dem ersten Faltungsfenster; und
Extrahieren einer dritten Reihe von Bilddaten aus dem zweiten Kanal in dem ersten Faltungsfenster.

4. Verfahren nach Anspruch 3, wobei das Extrahieren von Bilddaten aus dem ersten Faltungsfenster unter Verwendung der ersten Datenverarbeitungseinheit ferner Folgendes umfasst:
Darstellen aller Bilddaten in dem ersten Faltungsfenster unter Verwendung eines eindimensionalen Vektors als Reaktion auf den Abschluss der Extraktion aller Bilddaten aus allen Kanälen in dem ersten Faltungsfenster, wobei eine Länge des eindimensionalen Vektors ein Produkt aus der Anzahl C der Kanäle in dem Bild (310, 510), der Anzahl R der Zeilen in jedem Faltungsfenster (511) und der Anzahl S der Spalten in jedem Faltungsfenster (511) ist.

5. Verfahren nach einem der Ansprüche 1-4, das ferner Folgendes umfasst:

Aufteilen des Faltungskerns in mehrere Blöcke in Spalten, wobei die mehreren Blöcke einen ersten Block und einen zweiten Block umfassen;
paralleles Transponieren mehrerer Datenspalten in dem ersten Block unter Verwendung der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529); und
Transponieren mehrerer Datenspalten in dem zweiten Block parallel unter Verwendung der mehreren Datenverarbeitungseinheiten (321, 322, 323, 491,493, 494, 499, 521, 522, 523, 529) als Reaktion auf den Abschluss der Transposition mehrerer Spalten von Daten in dem ersten Block;
wobei das parallele Transponieren mehrerer Datenspalten in dem ersten Block unter Verwendung der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) vorzugsweise Folgendes umfasst:

Transponieren einer ersten Datenspalte in dem ersten Block unter Verwendung einer ersten Datenverarbeitungseinheit der verfügbaren Datenverarbeitungseinheiten; und
Transponieren einer zweiten Datenspalte im zweiten Block unter Verwendung einer zweiten Datenverarbeitungseinheit der verfügbaren Datenverarbeitungseinheiten.

6. Verfahren nach Anspruch 5, wobei die Unterteilung des Faltungskerns in mehrere Blöcke in Spalten Folgendes umfasst:
Aufteilen des Faltungskerns in die mehreren Blöcke basierend auf einer Anzahl der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529).

7. Beschleunigungsvorrichtung zum Falten von Bildern, die mehrere Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) enthält, die dazu konfiguriert sind, Bilddaten parallel aus mehreren Faltungsfenstern zu extrahieren, um eine Bildfaltungsoperation in eine Multiplikation von zweidimensionalen Matrizen zu transformieren, wobei die Beschleunigungsvorrichtung ferner Folgendes umfasst:

ein Faltungsfenstergruppenteilungsmodul (710), das dazu konfiguriert ist, ein Bild (310, 510) in N Faltungsfenster zu teilen, die in mehrere Gruppen von Faltungsfenstern gruppiert sind, wobei jede Gruppe von Faltungsfenstern eine Anzahl (P) von Faltungsfenstern enthält, die der Anzahl von verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) entspricht;
ein erstes konkurrierendes Extraktionsmodul (720), das dazu konfiguriert ist, Bilddaten parallel aus den (P) Faltungsfenstern in einer ersten Gruppe von Faltungsfenstern der mehreren Gruppen von Faltungsfenstern unter Verwendung der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 521, 522, 523, 529) zu extrahieren,
wobei jede Verarbeitungseinheit (321, 322, 323, 491, 493, 521, 522, (321, 322, 323, 491, 493, 521, 522, 5223, 529) Bilddaten aus einem entsprechenden Faltungsfenster extrahiert, einschließlich der Darstellung der entsprechenden Bilddaten durch Verwendung eines eindimensionalen Vektors mit einer Länge CxRxS, wobei C eine Anzahl von Kanälen in dem Bild (310, 510) ist, R eine Anzahl von Zeilen in jedem Faltungsfenster (511)

ist, und S eine Anzahl von Spalten in jedem Faltungsfenster (511) ist;
ein zweites konkurrierendes Extraktionsmodul (730), das dazu konfiguriert ist, in Reaktion auf den Abschluss der Extraktion von Bilddaten aus der ersten Gruppe von Faltungsfenstern Bilddaten parallel aus den (P) Faltungsfenstern in einer zweiten Gruppe von Faltungsfenstern der mehreren Gruppen von Faltungsfenstern unter Verwendung der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 521, 522, 523, 529) zu extrahieren und das Extrahieren der Bilddaten zu wiederholen, bis die Bilddaten in allen N Faltungsfenstern des Bildes (310, 510) extrahiert sind; und
ein Datenspeichermodul, das dazu konfiguriert ist, alle Bilddaten in den mehreren Gruppen von Faltungsfenstern in einem Zielspeicher unter Verwendung einer zweidimensionalen Matrix mit N Zeilen und CxRxS Spalten zu speichern, wobei N eine Anzahl aller Faltungsfenster in den mehreren Gruppen von Faltungsfenstern ist,
wobei die Vorrichtung dazu konfiguriert ist, einen Faltungskern, der durch eine zweidimensionale Matrix mit F Zeilen und CxRxS Spalten dargestellt wird, in eine zweidimensionale Matrix mit CxRxS Zeilen und F Spalten zu transponieren, so dass die Bildfaltungsoperation in eine Multiplikation der beiden zweidimensionalen Matrizen umgewandelt wird.

8. Vorrichtung nach Anspruch 7, wobei die verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 521, 522, 523, 529) eine erste Datenverarbeitungseinheit und eine zweite Datenverarbeitungseinheit umfassen, die erste Gruppe von Faltungsfenstern ein erstes Faltungsfenster und ein zweites Faltungsfenster umfasst, und das erste konkurrierende Extraktionsmodul Folgendes umfasst:

ein erstes Datenextraktionsmodul, das dazu konfiguriert ist, Bilddaten aus dem ersten Faltungsfenster unter Verwendung der ersten Datenverarbeitungseinheit zu extrahieren; und
ein zweites Datenextraktionsmodul, das dazu konfiguriert ist, Bilddaten aus dem zweiten Faltungsfenster unter Verwendung der zweiten Datenverarbeitungseinheit zu extrahieren.

9. Vorrichtung nach Anspruch 8, wobei das erste Datenextraktionsmodul Folgendes umfasst:

ein erstes Extraktionsmodul, das dazu konfiguriert ist, eine erste Zeile von Bilddaten aus einem ersten Kanal in dem ersten Faltungsfenster zu extrahieren;
ein zweites Extraktionsmodul, das dazu konfiguriert ist, eine zweite Zeile von Bilddaten aus dem ersten Kanal in dem ersten Faltungsfenster zu extrahieren; und
ein drittes Extraktionsmodul, das dazu konfiguriert ist, eine dritte Zeile von Bilddaten aus dem ersten Kanal in dem ersten Faltungsfenster zu extrahieren;
wobei das erste Datenextraktionsmodul vorzugsweise weiterhin Folgendes umfasst:
ein zweites Kanalextraktionsmodul, das dazu konfiguriert ist, als Reaktion auf den Abschluss der Extraktion aller Bilddaten aus dem ersten Kanal in dem ersten Faltungsfenster:

eine erste Zeile von Bilddaten aus einem zweiten Kanal in dem ersten Faltungsfenster zu extrahieren;
eine zweite Zeile von Bilddaten aus dem zweiten Kanal in dem ersten Faltungsfenster zu extrahieren; und
eine dritte Zeile von Bilddaten aus dem zweiten Kanal in dem ersten Faltungsfenster zu extrahieren.

10. Vorrichtung nach Anspruch 9, wobei das erste Datenextraktionsmodul ferner Folgendes umfasst:
ein Datendarstellungsmodul, das dazu konfiguriert ist, als Reaktion auf den Abschluss der Extraktion aller Bilddaten aus allen Kanälen in dem ersten Faltungsfenster alle Bilddaten in dem ersten Faltungsfenster unter Verwendung eines eindimensionalen Vektors darzustellen, wobei eine Länge des eindimensionalen Vektors ein Produkt aus der Anzahl C von Kanälen in dem Bild (310, 510), der Anzahl R von Zeilen in jedem Faltungsfenster (511) und der Anzahl C von Spalten in jedem Faltungsfenster (511) ist.

11. Vorrichtung nach einem der Ansprüche 7-10, die ferner Folgendes umfasst:

ein Blockteilungsmodul, das dazu konfiguriert ist, eine Matrix in mehrere Blöcke in Spalten zu unterteilen, wobei die mehreren Blöcke einen ersten Block und einen zweiten Block umfassen;
ein erstes konkurrierendes Transpositionsmodul, das dazu konfiguriert ist, mehrere Datenspalten im ersten Block unter Verwendung der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 521, 522, 523, 529) parallel zu transponieren; und
ein zweites konkurrierendes Transpositionsmodul, das dazu konfiguriert ist, als Reaktion auf den Abschluss der Transposition mehrerer Datenspalten im ersten Block mehrere Datenspalten im zweiten Block durch Verwendung der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 521, 522, 523, 529) parallel

zu transponieren;
wobei das erste gleichzeitige Transpositionsmodul vorzugsweise Folgendes umfasst:

ein erstes Matrixtranspositionsmodul, das dazu konfiguriert ist, eine erste Datenspalte in dem ersten Block unter Verwendung der ersten Datenverarbeitungseinheit der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 521, 522, 523, 529) zu transponieren; und
ein zweites Matrixtranspositionsmodul, das dazu konfiguriert ist, eine zweite Datenspalte in dem zweiten Block unter Verwendung der zweiten Datenverarbeitungseinheit der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 521, 522, 523, 529) zu transponieren.

12. Vorrichtung nach Anspruch 11, wobei das Blockteilungsmodul Folgendes umfasst:
ein zweites Blockteilungsmodul, das dazu konfiguriert ist, die Matrix auf der Grundlage einer Anzahl der verfügbaren Datenverarbeitungseinheiten (321, 322, 323, 491, 493, 521, 522, 523, 529) in mehrere Blöcke zu unterteilen.

13. Computerlesbares Speichermedium, auf dem ein Computerprogramm gespeichert ist, das bei Ausführung des Programms durch einen Prozessor das Verfahren nach einem der Ansprüche 1 bis 6 implementiert.

Revendications

1. Procédé d'extraction de données d'image, en parallèle, de multiples fenêtres de convolution, par un dispositif accélérateur incluant de multiples unités de traitement de données (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529), pour transformer une opération de convolution d'image en une multiplication de matrices bidimensionnelles, le procédé comprenant les étapes ci-dessous consistant à :

diviser (202) une image (310, 510) en N fenêtres de convolution regroupées en de multiples groupes de fenêtres de convolution, dans lequel chaque groupe de fenêtres de convolution inclut un nombre (P) de fenêtres de convolution correspondant au nombre d'unités de traitement de données disponibles (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) ;
extraire (204) des données d'image, en parallèle, des (P) fenêtres de convolution (311, 312, 313, 511), dans un premier groupe de fenêtres de convolution des multiples groupes de fenêtres de convolution, en utilisant les unités de traitement de données disponibles (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529), dans lequel l'étape d'extraction de données d'image, d'une fenêtre de convolution (511), par une unité de traitement correspondante (521, 522, 523, 529), comprend l'étape consistant à représenter des données d'image correspondantes en utilisant un vecteur unidimensionnel ayant une longueur de C x R x S, dans lequel « C » est un nombre de canaux dans l'image (310, 510), « R » est un nombre de lignes dans chaque fenêtre de convolution (511), et « S » est un nombre de colonnes dans chaque fenêtre de convolution (511) ;
extraire (206), en réponse à l'achèvement de l'extraction de données d'image à partir du premier groupe de fenêtres de convolution, des données d'image, en parallèle, des (P) fenêtres de convolution, dans un second groupe de fenêtres de convolution des multiples groupes de fenêtres de convolution, en utilisant les unités de traitement de données disponibles (321, 322, 323, 491, 493, 521, 522, 523, 529) ;
dans lequel l'étape d'extraction (206) des données d'image est répétée jusqu'à ce que les données d'image dans la totalité des N fenêtres de convolution de l'image (310, 510) soient extraites ; et
stocker toutes les données d'image extraites dans les multiples groupes de fenêtres de convolution, dans une mémoire cible, en utilisant une matrice bidimensionnelle présentant N lignes et C $\times$ R $\times$ S colonnes, dans lequel un noyau de convolution représenté par une matrice bidimensionnelle présentant F lignes et C $\times$ R $\times$ S colonnes est transposé dans une matrice bidimensionnelle présentant C $\times$ R $\times$ S lignes et F colonnes, de sorte qu'une opération de convolution d'image est transformée en une multiplication des deux matrices bidimensionnelles.

2. Procédé selon la revendication 1, dans lequel les unités de traitement de données disponibles (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) comprennent une première unité de traitement de données et une seconde unité de traitement de données, le premier groupe de fenêtres de convolution comprend une première fenêtre de convolution et une seconde fenêtre de convolution, et l'étape d'extraction (204) de données d'image, en parallèle, à partir des (P) fenêtres de convolution (311, 312, 313, 511), dans le premier groupe de fenêtres de convolution, en utilisant les multiples unités de traitement de données (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529), comprend les étapes ci-dessous consistant à :

extraire des données d'image de la première fenêtre de convolution, en utilisant la première unité de traitement de données ; et
extraire des données d'image de la seconde fenêtre de convolution, en utilisant la seconde unité de traitement de données.

3. Procédé selon la revendication 2, dans lequel l'étape d'extraction des données d'image, de la première fenêtre de convolution, en utilisant la première unité de traitement de données, comprend les étapes ci-dessous consistant à :

extraire une première ligne de données d'image, d'un premier canal, dans la première fenêtre de convolution ;
extraire une deuxième ligne de données d'image, du premier canal, dans la première fenêtre de convolution ; et
extraire une troisième ligne de données d'image, du premier canal, dans la première fenêtre de convolution ;
dans lequel, de préférence, l'étape d'extraction de données d'image, de la première fenêtre de convolution, en utilisant la première unité de traitement de données,
comprend en outre les étapes ci-dessous consistant à :
en réponse à l'achèvement de l'extraction de la totalité des données d'image, du premier canal, dans la première fenêtre de convolution :

extraire une première ligne de données d'image, d'un second canal, dans la première fenêtre de convolution ;
extraire une deuxième ligne de données d'image, du second canal, dans la première fenêtre de convolution ; et
extraire une troisième ligne de données d'image, du second canal, dans la première fenêtre de convolution.

4. Procédé selon la revendication 3, dans lequel l'étape d'extraction de données d'image, de la première fenêtre de convolution, en utilisant la première unité de traitement de données, comprend en outre l'étape ci-dessous consistant à :
représenter, en réponse à l'achèvement de l'extraction de la totalité des données d'image, de tous les canaux, dans la première fenêtre de convolution, toutes les données d'image dans la première fenêtre de convolution, en utilisant un vecteur unidimensionnel, une longueur du vecteur unidimensionnel étant un produit du nombre C de canaux dans l'image (310, 510), du nombre R de lignes dans chaque fenêtre de convolution (511), et du nombre S de colonnes dans chaque fenêtre de convolution (511).

5. Procédé selon l'une quelconque des revendications 1 à 4, comprenant en outre les étapes ci-dessous consistant à :

diviser le noyau de convolution en de multiples blocs en colonnes, les multiples blocs comprenant un premier bloc et un second bloc ;
transposer de multiples colonnes de données dans le premier bloc, en parallèle, en utilisant les unités de traitement de données disponibles (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) ; et
transposer, en réponse à l'achèvement de la transposition de multiples colonnes de données dans le premier bloc, de multiples colonnes de données dans le second bloc, en parallèle, en utilisant les unités de traitement de données disponibles (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) ;
dans lequel, de préférence, l'étape de transposition de multiples colonnes de données dans le premier bloc, en parallèle, en utilisant les multiples unités de traitement de données (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529), comprend les étapes ci-dessous consistant à :

transposer une première colonne de données dans le premier bloc, en utilisant une première unité de traitement de données parmi les unités de traitement de données disponibles ; et
transposer une second colonne de données dans le second bloc, en utilisant une seconde unité de traitement de données parmi les unités de traitement de données disponibles.

6. Procédé selon la revendication 5, dans lequel l'étape de division du noyau de convolution en de multiples blocs en colonnes, comprend l'étape ci-dessous consistant à :

diviser le noyau de convolution en les multiples blocs, sur la base d'un nombre des unités de traitement de données disponibles (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529).

7. Dispositif accélérateur pour une convolution d'image incluant de multiples unités de traitement de données (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) configurées de manière à extraire des données d'image, en

parallèle, de multiples fenêtres de convolution, pour transformer une opération de convolution d'image en une multiplication de matrices bidimensionnelles, le dispositif accélérateur comprenant en outre :

un module de division en groupes de fenêtres de convolution (710), configuré de manière à diviser une image (310, 510) en N fenêtres de convolution regroupées en de multiples groupes de fenêtres de convolution, dans lequel chaque groupe de fenêtres de convolution inclut un nombre (P) de fenêtres de convolution correspondant au nombre d'unités de traitement de données disponibles (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529) ;
un premier module d'extraction simultanée (720), configuré de manière à extraire des données d'image, en parallèle, des (P) fenêtres de convolution (311, 312, 313, 511), dans un premier groupe de fenêtres de convolution des multiples groupes de fenêtres de convolution, en utilisant les unités de traitement de données disponibles (321, 322, 323, 491, 493, 494, 499, 521, 522, 523, 529), dans lequel chaque unité de traitement (321, 322, 323, 491, 493, 521, 522, 523, 529) extrait des données d'image, d'une fenêtre de convolution correspondante, ce qui inclut de représenter les données d'image correspondantes en utilisant un vecteur unidimensionnel ayant une longueur de $C \times R \times S$, dans lequel « C » est un nombre de canaux dans l'image (310, 510), « R » est un nombre de lignes dans chaque fenêtre de convolution (511), et « S » est un nombre de colonnes dans chaque fenêtre de convolution (511);
un second module d'extraction simultanée (730), configuré de manière à extraire, en réponse à l'achèvement de l'extraction de données d'image, du premier groupe de fenêtres de convolution, des données d'image, en parallèle, des (P) fenêtres de convolution, dans un second groupe de fenêtres de convolution des multiples groupes de fenêtres de convolution, en utilisant les unités de traitement de données disponibles (321, 322, 323, 491, 493, 521, 522, 523, 529), et à répéter l'étape d'extraction des données d'image jusqu'à ce que les données d'image dans la totalité des N fenêtres de convolution de l'image (310, 510) soient extraites ; et
un module de stockage de données, configuré de manière à stocker toutes les données d'image dans les multiples groupes de fenêtres de convolution, dans une mémoire cible, en utilisant une matrice bidimensionnelle présentant N lignes et $C \times R \times S$ colonnes, dans lequel N est un nombre de la totalité des fenêtres de convolution dans les multiples groupes de fenêtres de convolution ;
dans lequel le dispositif accélérateur est en outre configuré de manière à transposer un noyau de convolution représenté par une matrice bidimensionnelle présentant F lignes et $C \times R \times S$ colonnes, en une matrice bidimensionnelle présentant $C \times R \times S$ lignes et F colonnes, de sorte qu'une opération de convolution d'image est transformée en une multiplication des deux matrices bidimensionnelles.

8. Dispositif accélérateur selon la revendication 7, dans lequel les unités de traitement de données disponibles (321, 322, 323, 491, 493, 521, 522, 523, 529) comprennent une première unité de traitement de données et une seconde unité de traitement de données, le premier groupe de fenêtres de convolution comprend une première fenêtre de convolution et une seconde fenêtre de convolution, et le premier module d'extraction simultanée comprend :

un premier module d'extraction de données, configuré de manière à extraire des données d'image, de la première fenêtre de convolution, en utilisant la première unité de traitement de données ; et
un second module d'extraction de données, configuré de manière à extraire des données d'image, de la seconde fenêtre de convolution, en utilisant la seconde unité de traitement de données.

9. Dispositif accélérateur selon la revendication 8, dans lequel le premier module d'extraction de données comprend :

un premier module d'extraction, configuré de manière à extraire une première ligne de données d'image, d'un premier canal, dans la première fenêtre de convolution ;
un deuxième module d'extraction, configuré de manière à extraire une deuxième ligne de données d'image, du premier canal, dans la première fenêtre de convolution ; et
un troisième module d'extraction, configuré de manière à extraire une troisième ligne de données d'image, du premier canal, dans la première fenêtre de convolution ;
dans lequel, de préférence, le premier module d'extraction de données comprend en outre :
un second module d'extraction de canal configuré de manière à, en réponse à l'achèvement de l'extraction de la totalité des données d'image, du premier canal, dans la première fenêtre de convolution :

extraire une première ligne de données d'image, d'un second canal, dans la première fenêtre de convolution ;
extraire une deuxième ligne de données d'image, du second canal, dans la première fenêtre de convolution ; et

extraire une troisième ligne de données d'image, du second canal, dans la première fenêtre de convolution.

10. Dispositif accélérateur selon la revendication 9, dans lequel le premier module d'extraction de données comprend en outre :
un module de représentation de données, configuré de manière à représenter, en réponse à l'achèvement de l'extraction de la totalité des données d'image, de tous les canaux, dans la première fenêtre de convolution, toutes les données d'image dans la première fenêtre de convolution, en utilisant un vecteur unidimensionnel, une longueur du vecteur unidimensionnel étant un produit du nombre C de canaux dans l'image (310, 510), du nombre R de lignes dans chaque fenêtre de convolution (511), et du nombre S de colonnes dans chaque fenêtre de convolution (511).

11. Dispositif accélérateur selon l'une quelconque des revendications 7 à 10, comprenant en outre :

un module de division en blocs, configuré de manière à diviser le noyau de convolution en de multiples blocs en colonnes, les multiples blocs comprenant un premier bloc et un second bloc ;
un premier module de transposition simultanée, configuré de manière à transposer de multiples colonnes de données dans le premier bloc, en parallèle, en utilisant les unités de traitement de données disponibles (321, 322, 323, 491, 493, 521, 522, 523, 529) ; et
un second module de transposition simultanée, configuré de manière à transposer, en réponse à l'achèvement de la transposition de multiples colonnes de données dans le premier bloc, de multiples colonnes de données dans le second bloc, en parallèle, en utilisant les unités de traitement de données disponibles (321, 322, 323, 491, 493, 521, 522, 523, 529) ;
dans lequel, de préférence, le premier module de transposition simultanée comprend :

un premier module de transposition de matrice, configuré de manière à transposer une première colonne de données dans le premier bloc, en utilisant la première unité de traitement de données parmi les unités de traitement de données disponibles (321, 322, 323, 491, 493, 521, 522, 523, 529) ; et
un second module de transposition de matrice, configuré de manière à transposer une seconde colonne de données dans le second bloc, en utilisant la seconde unité de traitement de données parmi les unités de traitement de données disponibles (321, 322, 323, 491, 493, 521, 522, 523, 529).

12. Dispositif accélérateur selon la revendication 11, dans lequel le module de division en blocs comprend :
un second module de division en blocs, configuré de manière à diviser la matrice en les multiples blocs, sur la base d'un nombre d'unités de traitement de données disponibles (321, 322, 323, 491, 493, 521, 522, 523, 529).

13. Support de stockage lisible par ordinateur, stockant un programme informatique sur celui-ci, dans lequel, lorsque le programme est exécuté par un processeur, le procédé selon l'une quelconque des revendications 1 à 6 est mis en œuvre.

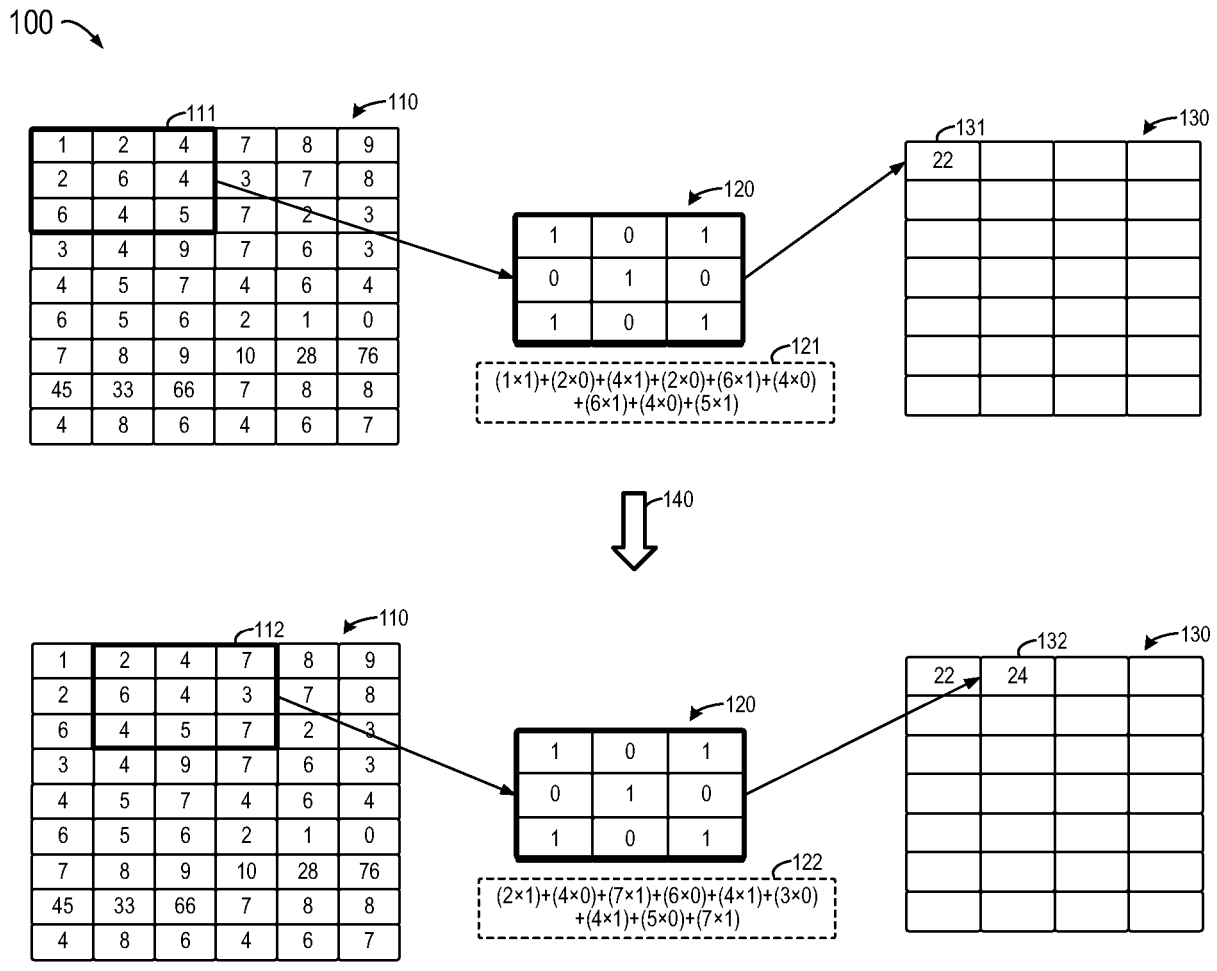
100
111
110
1 2 4 7 8 9
2 6 4 3 7 8
6 4 5 7 2 3
3 4 9 7 6 3
4 5 7 4 6 4
6 5 6 2 1 0
7 8 9 10 28 76
45 33 66 7 8 8
4 8 6 4 6 7
120
1 0 1
0 1 0
1 0 1
121
(1×1)+(2×0)+(4×1)+(2×0)+(6×1)+(4×0)
+(6×1)+(4×0)+(5×1)
131
130
22
140
112
110
1 2 4 7 8 9
2 6 4 3 7 8
6 4 5 7 2 3
3 4 9 7 6 3
4 5 7 4 6 4
6 5 6 2 1 0
7 8 9 10 28 76
45 33 66 7 8 8
4 8 6 4 6 7
120
1 0 1
0 1 0
1 0 1
122
(2×1)+(4×0)+(7×1)+(6×0)+(4×1)+(3×0)
+(4×1)+(5×0)+(7×1)
132
130
22
24

Fig. 1

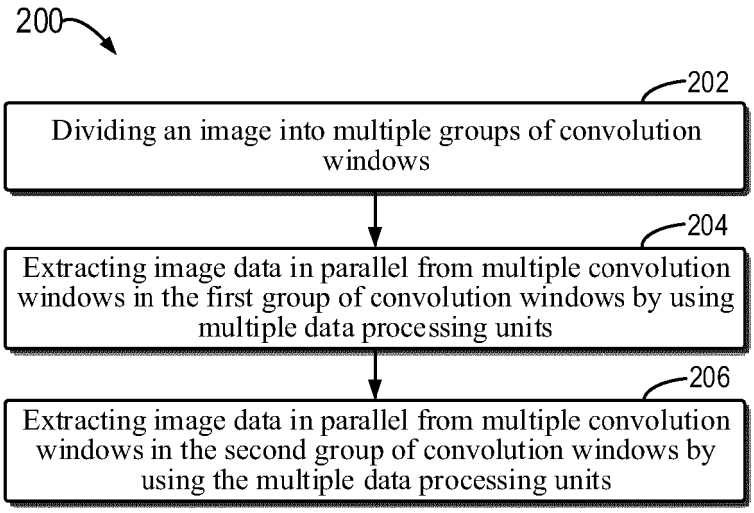
200
202
Dividing an image into multiple groups of convolution windows
204
Extracting image data in parallel from multiple convolution windows in the first group of convolution windows by using multiple data processing units
206
Extracting image data in parallel from multiple convolution windows in the second group of convolution windows by using the multiple data processing units

Fig. 2

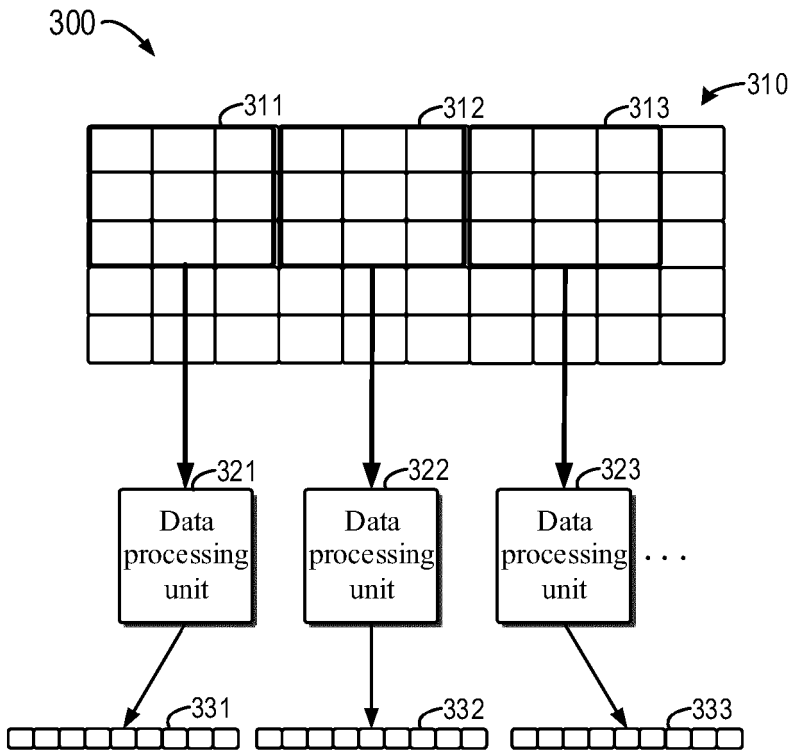
300
310
311
312
313
321
322
323
Data processing unit
Data processing unit
Data processing unit
331
332
333

Fig. 3

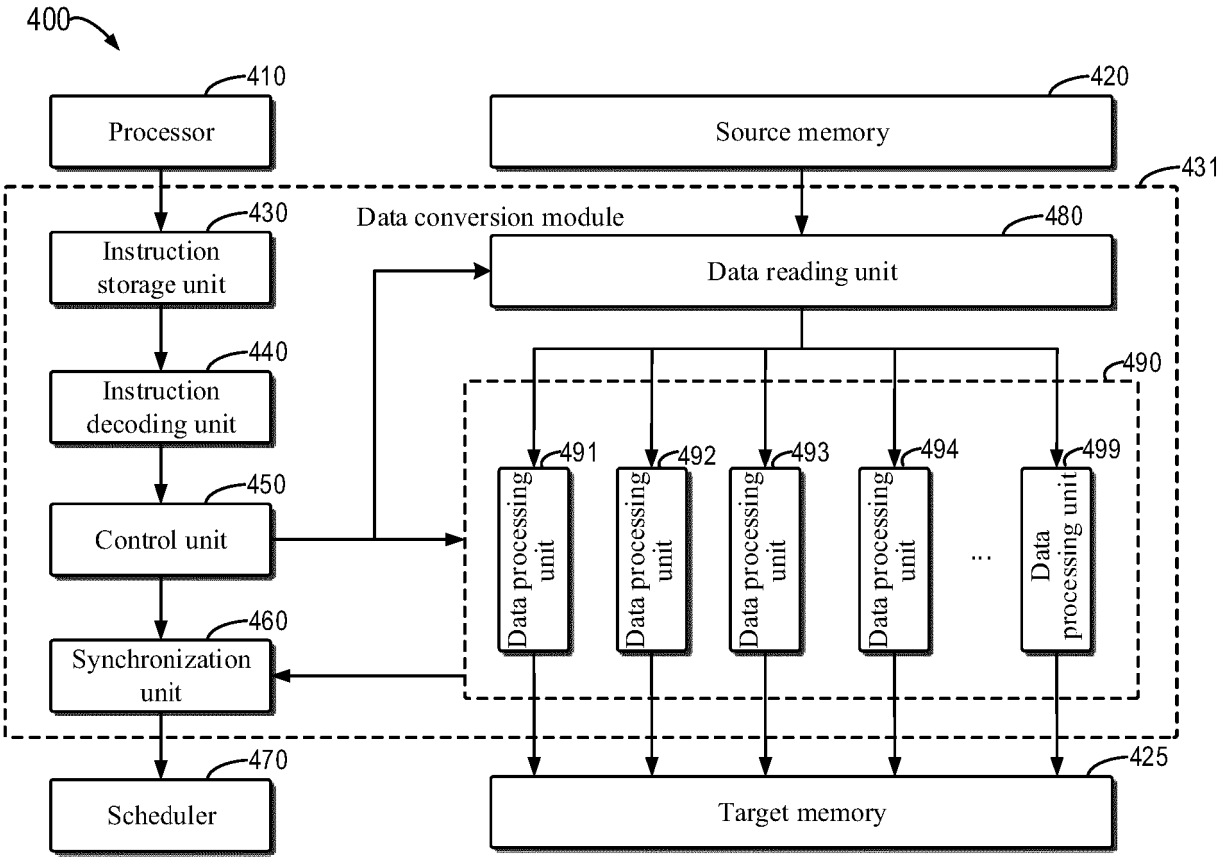
400
410
Processor
420
Source memory
431
Data conversion module
430
Instruction storage unit
480
Data reading unit
440
Instruction decoding unit
490
450
Control unit
491
492
493
494
499
Data processing unit
460
Synchronization unit
470
Scheduler
425
Target memory

Fig. 4

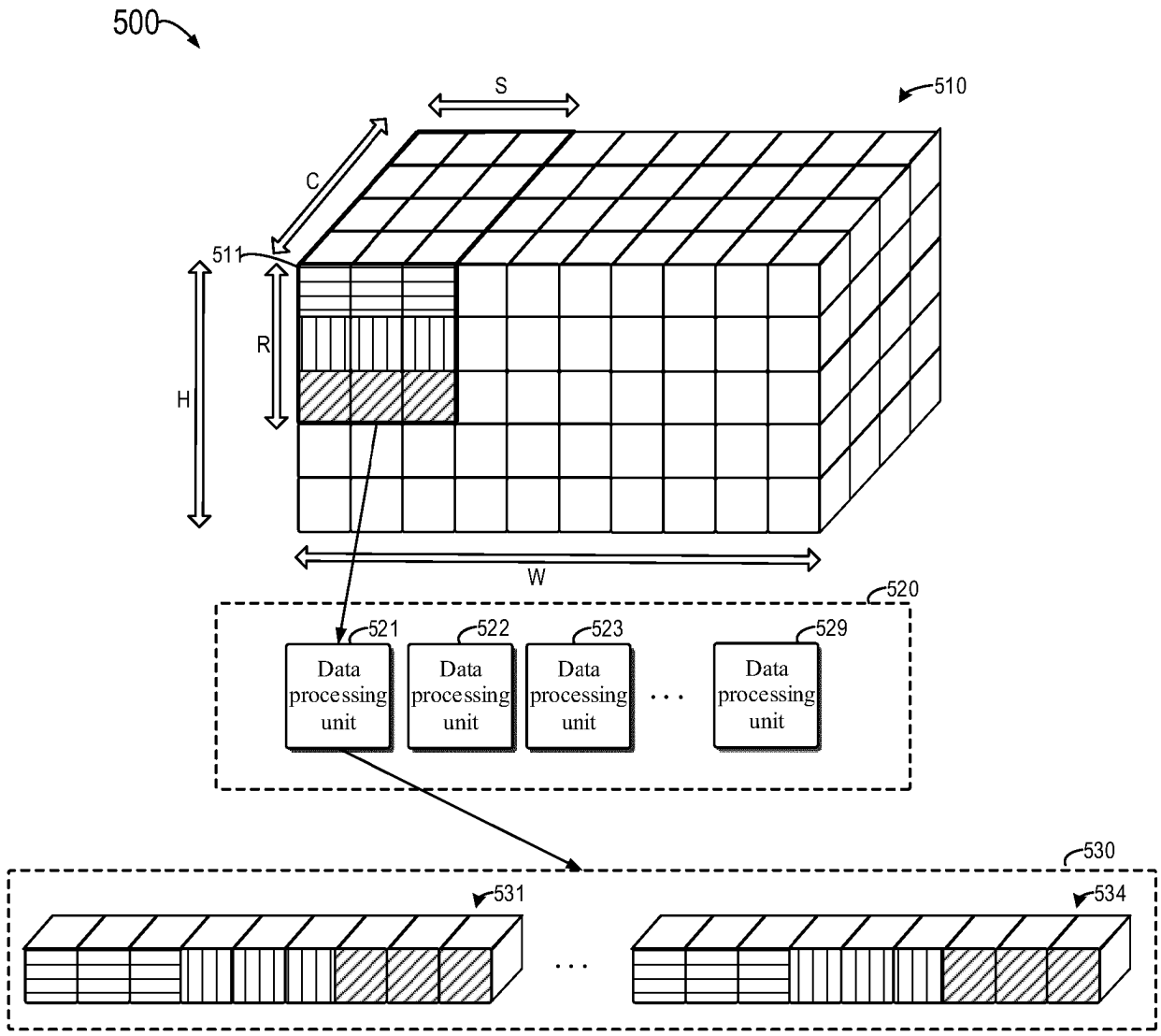
500
S
510
C
511
R
H
W
520
521
522
523
529
Data processing unit
Data processing unit
Data processing unit
Data processing unit
530
531
534

Fig. 5

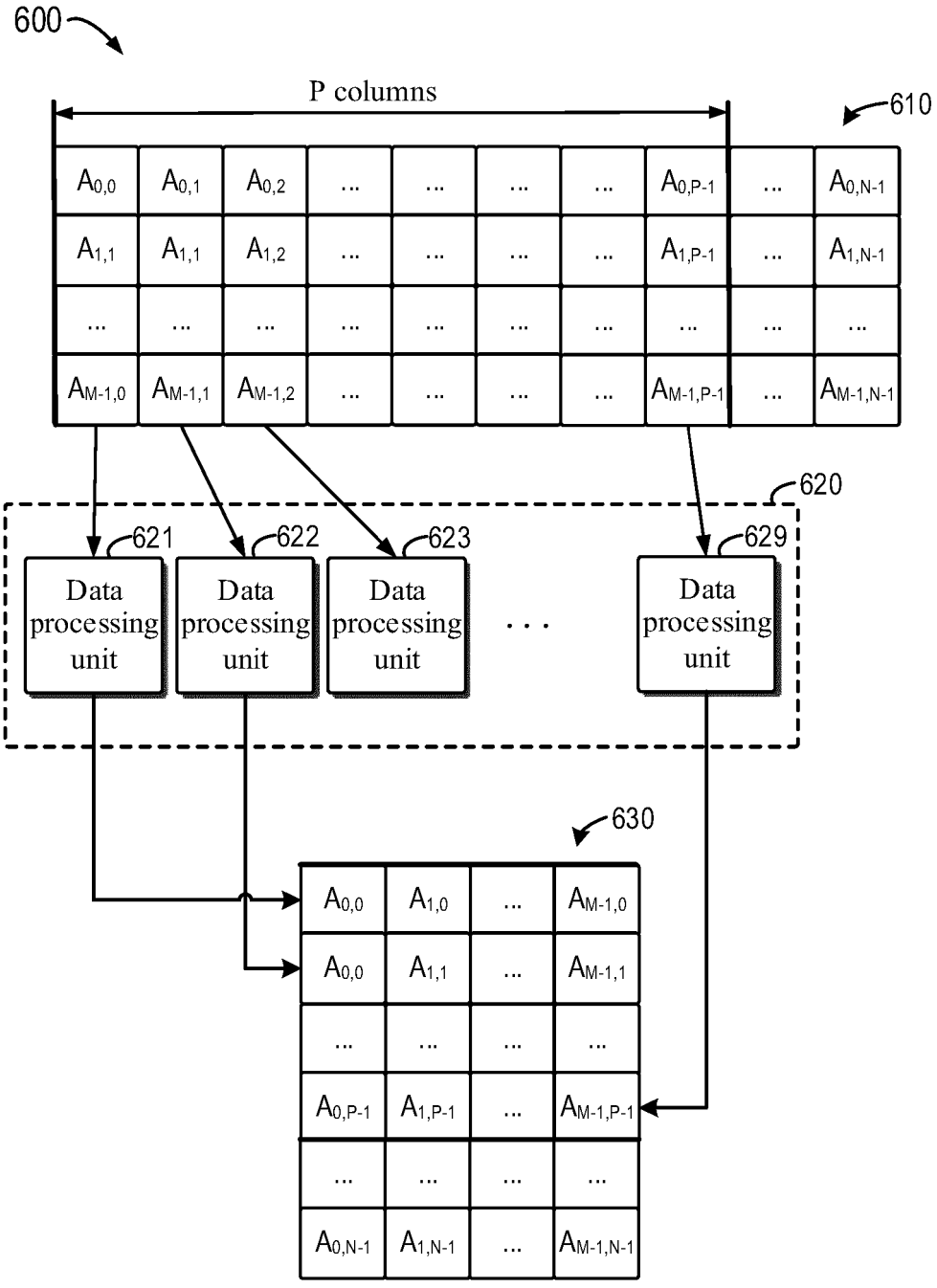
600
P columns
610
A0,0 A0,1 A0,2 ... ... A0,P-1 ... A0,N-1
A1,1 A1,1 A1,2 ... ... A1,P-1 ... A1,N-1
... ...
AM-1,0 AM-1,1 AM-1,2 ... ... AM-1,P-1 ... AM-1,N-1
620
621
622
623
629
Data processing unit
Data processing unit
Data processing unit
Data processing unit
630
A0,0 A1,0 ... AM-1,0
A0,0 A1,1 ... AM-1,1
... ...
A0,P-1 A1,P-1 ... AM-1,P-1
... ...
A0,N-1 A1,N-1 ... AM-1,N-1

Fig. 6

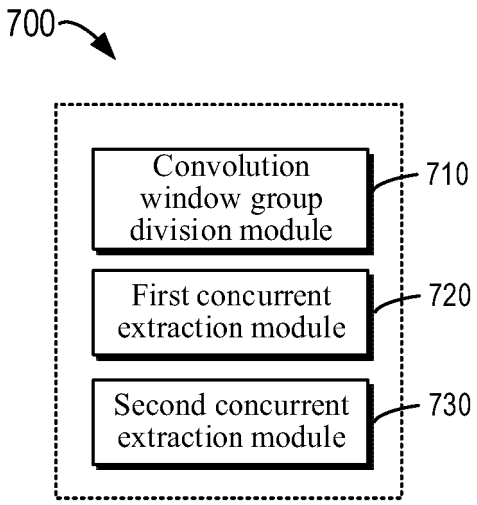
700
Convolution window group division module
710
First concurrent extraction module
720
Second concurrent extraction module
730

Fig. 7

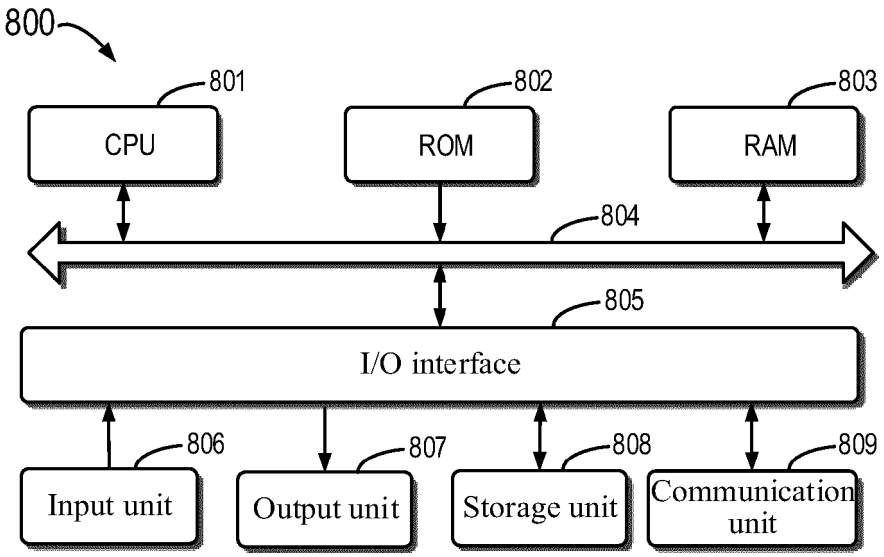
800
801
CPU
802
ROM
803
RAM
804
805
I/O interface
806
Input unit
807
Output unit
808
Storage unit
809
Communication unit

Fig. 8

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO 2019109795 A1 **[0005]**

Non-patent literature cited in the description

- **IJZERMAN et al.** AivoTTA. *Embedded Computer Systems, ACM, 2 Penn Plaza, Suite 701, New York, USA,* 15 July 2018, ISBN 978-1-4503-6494-2, 28-37 **[0006]**
- **IJZERMAN J. G.** Customized low power processor for object recognition a programmable high performance low power TTA-SIMD accelerator for CNN-based object recognition. *Master's thesis,* 31 December 2016 **[0006]**