

AN IMAGE PROCESSING METHOD AND APPARATUS

Technical field

The present application relates to the field of image process, and in particular, to an image processing method and an image processing apparatus.

Background

A neural network model is an operation model including a large number of nodes (or neurons) connected to each other. A commonly used neural network model includes an input layer, an output layer, and multiple hidden layers (also called hidden layers). As a neural network becomes larger and a structure of the neural network becomes more complex, hardware resources required for prediction and training are gradually increasing, and a deep learning neural network model can generally be run on a server with a high computing capability. A lightweight neural network model makes the size smaller and the speed faster while accuracy of the model is ensured, making it possible to run the neural network model in a mobile network.

An inverted residual block (Inverted residual block, IR block) dominates the mobile network, and the classic residual bottleneck is changed by introducing two design rules: inverted residual learning and the use of linear bottlenecks. The connection between bottlenecks will inevitably bring about information loss and gradient confusion, and the propagation of gradients between layers is weakened. Consequently, the convergence of training and the performance of the model are affected.

Summary

Embodiments of the present application provide an image processing method, an image processing apparatus, an image processing device and a computer-readable storage medium.

According to a first aspect, an embodiment of the present application provides an image processing method based on a neural network, and the method includes:

obtaining an image;

processing the image into at least one layer of image data on an input layer of the neural network model;

inputting the image data into a hidden layer of the neural network model for processing to obtain local high-dimensional feature data of the image data, wherein the hidden layer of the neural network model includes a plurality of processing units, and at least one or more processing units among the plurality of processing units are interconnected and connected to output of the input layer; and

outputting a feature map of the image, wherein the feature map of the image is obtained by pooling and fusing local high-dimensional features of the image data; where

in at least one processing unit, the high-dimensional feature data of the image as input of the at least one processing unit is undergone a shortcut connection process which sequentially include a first depthwise separable convolutional layer, a first pointwise convolutional layer, a second pointwise convolutional layer, and a second depthwise separable convolutional layer, and the local high-dimensional feature data of the image is outputted.

Compared with the image processing method in the prior art, In the process of image processing the present application is capable of:

- (1) alleviate information loss and gradient confusion effectively.
- (2) reduce computational costs and remove of redundant parameters from the neural network effectively.
- (3) reduce computational complexity and improve classification accuracy of images.
- (4) the internal structure of the computer system can be more streamlined, allowing image processing to be applied in lightweight neural networks such as mobile devices.

In a possible implementation of the first aspect, the method further includes that the first pointwise convolutional layer performs channel reduction on the image data, and the second pointwise convolutional layer performs channel expansion on the image data outputted by the

first pointwise convolutional layer. Not only parameters and calculation costs are reduced, but also a shortcut connection instead of a bottleneck connection can be used to connect a representation with a large number of channels.

In a possible implementation of the first aspect, the method further includes that no other convolution operations are performed between the first pointwise convolutional layer and the second pointwise convolutional layer. This structure greatly reduces performance by more than 1%.

In a possible implementation of the first aspect, the method further includes that the shortcut connection is established in a high-dimensional feature space of the image data, and a local feature tensor of the image data is extracted, to obtain the local high-dimensional feature data of the image data.

In a possible implementation of the first aspect, the method further includes that a linear activation function is used for data outputted by the first depthwise separable convolutional layer; a nonlinear activation function is used for data outputted by the first pointwise convolutional layer; a linear activation function is used for data outputted by the second pointwise convolutional layer; and a nonlinear activation function is used for data outputted by the second depthwise convolutional layer. In this way, more information can be passed from an input tensor to an output tensor, and more gradients are allowed to propagate across multiple layers.

In a possible implementation of the first aspect, the method further includes that the neural network model includes an input layer, multiple hidden layers, and an output layer, the hidden layer further includes a first convolutional layer, a fully convolutional layer, and a global average pooling layer, the fully convolutional layer includes at least one processing unit, the input image data is processed by the first convolutional layer including 32 output channels to obtain 32 feature maps of the image data. The feature map of the image data is processed by the fully convolutional layer to obtain the local high-dimensional feature data of the image data. The local high-dimensional feature data of the image data is processed by the global average pooling layer and is pooled as the feature map of the image.

According to a second aspect, an embodiment of the present application provides an image processing apparatus, including:

obtaining module, obtaining an image;

processing module, processing the image into at least one layer of image data based a neural network model; inputting the image data for processing to obtain local high-dimensional feature data of the image data, wherein the processing module comprises a plurality of processing units, and at least one or more processing units among the plurality of processing units are interconnected and connected to the outputting module; and

outputting module, outputting a feature map of the image, wherein the feature map of the image is obtained by pooling and fusing local high-dimensional features of the image data; wherein

in at least one processing unit, the high-dimensional feature data of the image as input of the at least one processing unit is undergone a shortcut connection process which sequentially comprise a first depthwise separable convolutional layer, a first pointwise convolutional layer, a second pointwise convolutional layer, and a second depthwise separable convolutional layer, and the local high-dimensional feature data of the image is outputted.

In a possible implementation of the second aspect, the apparatus further includes that the first pointwise convolutional layer performs channel reduction on the image data, and the second pointwise convolutional layer performs channel expansion on the image data outputted by the first pointwise convolutional layer. Not only parameters and calculation costs are reduced, but also a shortcut connection instead of a bottleneck connection can be used to connect a representation with a large number of channels.

In a possible implementation of the second aspect, the apparatus further includes that no other convolution operations are performed between the first pointwise convolutional layer and the second pointwise convolutional layer. This structure greatly reduces performance by more than 1%.

In a possible implementation of the second aspect, the apparatus further includes that the shortcut connection is established in a high-dimensional feature space of the image data, and a local feature tensor of the image data is extracted, to obtain the local high-dimensional feature data of the image data.

In a possible implementation of the second aspect, the apparatus further includes that a linear activation function is used for data outputted by the first depthwise separable convolutional layer; a nonlinear activation function is used for data outputted by the first pointwise convolutional layer; a linear activation function is used for data outputted by the

second pointwise convolutional layer; and a nonlinear activation function is used for data outputted by the second depthwise convolutional layer. In this way, more information can be passed from an input tensor to an output tensor, and more gradients are allowed to propagate across multiple layers.

In a possible implementation of the second aspect, the apparatus further includes that the neural network model includes an input layer, multiple hidden layers, and an output layer, the hidden layer further includes a first convolutional layer, a fully convolutional layer, and a global average pooling layer, the fully convolutional layer includes at least one processing unit, the input image data is processed by the first convolutional layer including 32 output channels to obtain 32 feature maps of the image data. The feature map of the image data is processed by the fully convolutional layer to obtain the local high-dimensional feature data of the image data. The local high-dimensional feature data of the image data is processed by the global average pooling layer and is pooled as the feature map of the image.

According to a third aspect, an embodiment of the present application provides an image processing device, including:

a memory, configured to store instructions executed by one or more processors of a system, and

a processor, being one of the processors of the system, and configured to execute the instructions to implement the image processing method, wherein:

obtaining an image;

processing the image into at least one layer of image data on an input layer of the neural network model;

inputting the image data into a hidden layer of the neural network model for processing, so as to obtain local high-dimensional feature data of the image data, wherein the hidden layer of the neural network model comprises a plurality of processing units, and at least one or more processing units among the plurality of processing units are interconnected and connected to output of the input layer; and

outputting a feature map of the image, wherein the feature map of the image is obtained by pooling and fusing local high-dimensional features of the image data; wherein

in at least one processing unit, the high-dimensional feature data of the image as input of the at least one processing unit is undergone a shortcut connection process which sequentially comprise a first depthwise separable convolutional layer, a first pointwise convolutional layer, a second pointwise convolutional layer, and a second depthwise separable convolutional layer, and the local high-dimensional feature data of the image is outputted.

In a possible implementation of the third aspect, the device further includes that the first pointwise convolutional layer performs channel reduction on the image data, and the second pointwise convolutional layer performs channel expansion on the image data outputted by the first pointwise convolutional layer.

In a possible implementation of the third aspect, the device further includes that no other convolution operations are performed between the first pointwise convolutional layer and the second pointwise convolutional layer.

In a possible implementation of the third aspect, the device further includes that the shortcut connection is established in a high-dimensional feature space of the image data, and a local feature tensor of the image data is extracted, to obtain the local high-dimensional feature data of the image data.

In a possible implementation of the third aspect, the device further includes that a linear activation function is used for data outputted by the first depthwise separable convolutional layer; a nonlinear activation function is used for data outputted by the first pointwise convolutional layer; a linear activation function is used for data outputted by the second pointwise convolutional layer; and a nonlinear activation function is used for data outputted by the second depthwise convolutional layer.

In a possible implementation of the third aspect, the device further includes that the neural network model comprises an input layer, multiple hidden layers, and an output layer, the hidden layer further comprises a first convolutional layer, a fully convolutional layer, and a global average pooling layer, the fully convolutional layer comprises at least one processing unit, the input image data is processed by the first convolutional layer comprising 32 output channels to obtain 32 output channels of feature maps of the image data; the feature map of the image data is processed by the fully convolutional layer to obtain the local high-dimensional feature data of the image data; the local high-dimensional feature data of the image data is processed by the global average pooling layer and is pooled as the feature map of the image.

According to a forth aspect, an embodiment of the present application provides a computer-readable storage medium encoded with a computer program, where instructions are stored on the computer-readable medium, and when the instructions are executed on a computer, the computer performs the image processing method according to any one possible implementations of the first aspect.

Brief description of the drawings

FIG. 1 is a schematic structural diagram of a neural network model according to some embodiments of the present application;

FIG. 2(a) is a schematic structural diagram of a classic bottleneck residual block according to some embodiments of the present application;

FIG. 2(b) is a schematic structural diagram of an inverted residual block according to some embodiments of the present application;

FIG. 2(c) is a schematic structural diagram of a processing unit according to some embodiments of the present application;

FIG. 3 is a structural block diagram of an electronic device according to some embodiments of the present application;

FIG. 4 is a flowchart of an image processing method according to some embodiments of the present application;

FIG. 5(a) is a network structural diagram of a bottleneck structure in which a standard convolution is replaced by a depthwise separable convolutions according to some embodiments of the present application;

FIG. 5(b) is a network structural diagram of a bottleneck structure in which a standard convolution is replaced by two depthwise separable convolutions according to some embodiments of the present application;

FIG. 5(c) is a network structural diagram of two pointwise convolution positions in a

switching processing unit according to some embodiments of the present application;

FIG. 5(d) is a network structural diagram of a processing unit according to some embodiments of the present application;

FIG. 6 is a diagram of comparison between a classic bottleneck structure with a depthwise separable convolution and a network structure of a processing unit according to some embodiments of the present application;

FIG. 7 is a schematic structural diagram of an image processing apparatus according to some embodiments of the present application;

FIG. 8(a) is a schematic diagram of a process of searching for a normal cell structure on a CIFAR-10 dataset by using DARTS according to some embodiments of the present application;

FIG. 8(b) is a schematic diagram of a process of searching for a reduced cell structure on a CIFAR-10 dataset by using DARTS according to some embodiments of the present application;

FIG. 9 is a schematic structural diagram of an electronic device for image processing according to some embodiments of the present application; and

FIG. 10 is a schematic structural diagram of an electronic device according to some embodiments of the present application.

Detailed description of the embodiments

Illustrative embodiments of the present application include but are not limited to an image processing method and device, and a medium.

It can be understood that, as used herein, the terms "module" and "unit" may refer to or include an application specific integrated circuit (ASIC), an electronic circuit, a processor (shared, dedicated, or group) and/or a memory that executes one or more software or firmware programs, a combinational logic circuit, and/or other suitable hardware components that provide the described functions, or may be a part of these hardware components.

It can be understood that, in the embodiments of the present application, the processor may be a microprocessor, a digital signal processor, a microcontroller, etc., and/or any combination thereof. According to another aspect, the processor may be a single-core processor, a multi-core processor, etc., and/or any combination thereof.

It can be understood that the image processing method provided in the present application can be implemented on various electronic devices, including but not limited to an electronic device such as a server, a distributed server cluster including multiple servers, a mobile phone, a tablet computer, a laptop computer, a desktop computer, and a wearable device, a head-mounted display, a mobile email device, a portable game console, a portable music player, a reader device, a personal digital assistant, a virtual reality or augmented reality device, or a television with one or more processors embedded or coupled.

In particular, the image processing method provided in the present application is suitable for an edge device. Edge computing is a distributed open platform (architecture) that integrates core capabilities of network, computing, storage, and application on the edge of the network near the source of objects or data, and can meet key requirements of real-time service, data optimization, application intelligence, security and privacy protection. For example, an edge device may be a device capable of performing edge calculation on video data at an end close to a video data source (network intelligent camera) in a video surveillance system.

As shown in Figure 1, according to some embodiments of the present application, a neural network model is provided, and is called an MobileNeXt model.

The neural network model shown in Figure 1 can be used for image processing. As shown in Figure 1, the neural network model consists of input layer, multiple implicit layers, and output layer for a total of n network layers. Of these, the first layer is called the input layer, the last layer is called the output layer, and the other layers are called the hidden layer. The input layer receives objects to be processed in different dimensions in the form of arrays, such as pictures, video frames, etc. The hidden layer consists of three types of underlying constructs: convolutional, pooling, and fully connected layers. In the basic construction, the convolutional and pooled layers are specific to convolutional neural networks. More complex algorithms include complex constructions such as Inception modules, residual blocks, etc. The convolutional cores in the convolutional layer contain weighting factors, while the pooling layer does not.

As shown in Figure 1, the structure of the underlying network model consists of 1 input layer, 3 convolutional layers, 2 pooling layers, 1 fully connected layer and 1 output layer. The order of the three types of foundation construction in the implicit layer is usually: input layer - convolutional layer - pooling layer - fully connected layer - output layer.

A specific process of processing the neural network model shown in figure 1 by using sample images when image processing is performed is described in detail below.

Before the neural network model is used to perform image processing, a large amount of sample image data and expected result data need to be inputted into the model for training. Image data of each sample image is inputted to n nodes of an input layer of the neural network model shown in figure 1. Calculation of a hidden layer is performed, and image processing result data is finally generated after calculation of an output layer. It should be noted that, when a large number of sample images are used to train the model, each complete training process corresponds to only one image. For example, there are a total of 1000 images, the first image can be trained first, after the training of the first image is completed, the second image is trained, and the images are trained by analog until the neural network model converges. After training of each image is completed, the processing result data finally outputted by the neural network model is compared with the expected result data, an error is calculated, and weights of nodes in other network layers are adjusted based on the error. In this way, the neural network model is trained by inputting the image data of the above image, and the weights are continuously adjusted. When the error between the image processing result data finally outputted by the neural network model and the expected result data is less than an error threshold, it is determined that the neural network model converges.

In some embodiments, the hidden layer further includes a first convolutional layer, a fully convolutional layer, and a global average pooling layer, and the fully convolutional layer includes at least one processing unit. Details of a network architecture of this neural network model are shown in Table 1. Each row represents a series of processing units, which repeat "b" times. A reduction rate used in each processing unit is represented by "t". A stride "s" of the first processing unit at each stage is set to be 2, and all other strides are set to be 1. Each convolutional layer is followed by a batch normalization layer. Kernel sizes of all spatial convolutions are set to be 3×3 and it is assumed that there are k categories in total. However, for a processing unit with different input and output channels, identity mapping is not added.

Table 1

No.	t	Output dimension	s	b	Input dimension	Operator
1	-	112×112×32	2	1	224×224×3	conv2D 3×3
2	2	56×56×96	2	1	112×112×32	Processing unit
3	6	56×56×144	1	1	56×56×96	Processing unit
4	6	28×28×192	2	3	56×56×144	Processing unit
5	6	14×14×288	2	3	28×28×192	Processing unit
6	6	14×14×384	1	4	14×14×288	Processing unit
7	6	7×7×576	2	4	14×14×384	Processing unit
8	6	7×7×960	1	2	7×7×576	Processing unit
9	6	7×7×1280	1	1	7×7×960	Processing unit
10	-	1×1×1280	-	1	7×7×1280	Avgpool 7×7
11	-	k	-	1	1×1×1280	conv2D 1×1

It can be understood that the above description of the technical solution for training the neural network model shown in figure 1 is merely exemplary and not limiting.

The mobile neural network model in the prior art for image processing is mostly composed of at least one stacked processing unit. The processing unit is designed based on a classic bottleneck (bottleneck) residual block or an inverted residual block. The processing unit in the MobileNeXt model proposed in the present application is different from the existing technical solution, and is described in detail below with reference to the drawings.

Figure 2(a) is a schematic structural diagram of a classic bottleneck residual block according to some embodiments of the present application. The bottleneck structure was first introduced in ResNet. As shown in figure 2(a), a typical bottleneck structure is composed of three convolutional layers: a 1×1 convolution for channel reduction, a 3×3 standard convolution for spatial feature extraction, and another 1×1 convolution for channel expansion. Channels in

each convolutional layer are expanded, and group convolution is applied to intermediate bottleneck convolution to gather richer feature representations, or an attention-based module is introduced to explicitly model interdependence between channels, so that the classic bottleneck structure can be further improved. However, due to complexity of the model, it is only applicable to a heavy-weight neural network, and it is difficult to be applied in a lightweight neural network.

To reduce the amount of calculation and improve efficiency of the neural network model, depthwise separable convolution is used to replace standard convolution in this field. Specifically, for example, a convolution with a weight of $k \times k \times M \times N$ can be decomposed into two convolutions, where $k \times k$ is a kernel size, and M and N are respectively the number of input channels and output channels. The first one convolution is a convolution operation of M channels in a depth direction of $k \times k$ (that is, channel convolution), to understand a spatial correlation between positions of the channels. The second convolution is a pointwise convolution, and linear combination of channels is learned to generate new features. The combination of the pointwise convolution and the $k \times k$ depthwise separable convolution has significant less parameters and a smaller amount of calculation. Therefore, the parameters and calculation costs can be significantly reduced by using depthwise separable convolution when the processing unit is constructed.

Figure 2(b) is a schematic structural diagram of an inverted residual block according to some embodiments of the present application. The inverted residual block is first introduced in MobileNetV2 and is specifically made for a mobile device, especially a device with limited computing resource budget. As shown in figure 2(b), different from the classic bottleneck structure, in order to reduce calculation amount, a low-dimensional compression tensor is used as input and is expanded to higher-dimensional space by using pointwise convolution. Then, a depthwise separable convolution is applied to perform spatial context encoding, and then another pointwise convolution is performed, to generate a low-dimensional feature tensor as input of a next residual block. The inverted residual block uses two unique architectural designs, namely establishing a shortcut connection between low-dimensional bottlenecks and using linear bottlenecks, to improve efficiency without causing excessive performance degradation. Compared with the classic MobileNet consisting of a 3×3 depthwise separable convolution, performance is improved, and complexity of the model is optimized.

The inverted residual block is widely used in image processing based on mobile network

architectures due to improvement of performance. For example, in ShuffleNetV2, a channel split module (channel split module) is inserted before an inverted residual block, and a channel shuffle module is added after the inverted residual block (channel shuffle module). In HBONet, a downsampling operation is introduced to the inverted residual block, to model more abundant spatial information. MobileNetV3 proposes to search for an optimal activation function and an expansion ratio of an inverted residual block at each stage. MixNet proposes to search for an optimal kernel size of a depthwise separable convolution in an inverted residual block. EfficientNet is also based on an inverted residual block, but the difference is that a scaling method is used to control network weights, including input resolution, network depth, and network width.

Although performing well, in the inverted residual block, the feature encoded by an intermediate expansion layer is mapped to the low-dimensional feature mapping, and sufficient useful information cannot be retained due to channel compression. In addition, a recent research shows that a wider architecture is more conducive to alleviating gradient confusion and improving the performance of the neural network. Due to low dimensionality expressed between adjacent inverted residual blocks, the shortcut connection placed between the bottlenecks may prevent the gradient of the top layer from successfully propagating to the bottom layer during model training. Unlike all the above methods, the classic bottleneck structure is improved in the present application, and it is proved that the performance of the classic bottleneck structure in a mobile environment is better than that of the inverted residual block.

In view of the limitations for image processing of the above inverted residual block, design rules of the inverted residual block are reconsidered in the present application, a processing unit is proposed, and the above problem can be solved by an operation of inverting inverted residuals. Figure 2(c) is a schematic structural diagram of a processing unit according to some embodiments of the application. As shown in figure 2(c), a bottleneck structure is used by the processing unit, which is similar to the inverted residual block and the classic bottleneck residual block, but the design principle and the internal structure are quite different. In the processing unit, the thickness of each unit is used to indicate the corresponding number of channels. It can be learned that compared with the inverted residual block, the processing unit reverses the idea of establishing a shortcut between the bottlenecks, and adds a depthwise separable convolution at both ends of a residual path, so that improved model performance in terms of classification accuracy and computational speed. Details are explained below.

Figure 3 is a structural block diagram of an electronic device 100 according to some embodiments of the present application. Specifically, as shown in figure 3, the electronic device 100 includes one or more processors 104, system control logic 108 connected to at least one of the processors 104, a system memory 112 connected to the system control logic 108, and a nonvolatile memory (NVM) 116 connected to the system control logic 108, and a network interface 120 connected to the system control logic 108.

In some embodiments, the processor 104 may include one or more single-core or multi-core processors. In some embodiments, the processor 104 may include any combination of a general-purpose processor and a dedicated processor (for example, a graphics processing unit, an application processor, and a baseband processor). In an embodiment in which the electronic device 100 uses an evolved NodeB (Evolved Node B, eNB) or radio access network (radio access network, RAN) controller, the processor 104 may be configured to perform various corresponding embodiments.

In some embodiments, the processor 104 may be configured to perform image processing based on a neural network model. Specifically, for example, in a process in which an image is processed by each layer of the neural network model, the processor 104 continuously adjusts the neural network model according to acquired image information until a highly accurate image feature map is obtained. The processor 104 may further periodically update the neural network model to better adapt to changes in various actual demands on the neural network model.

In some embodiments, the system control logic 108 may include any suitable interface controller to provide any suitable interface for at least one of the processors 104 and/or any suitable device or component in communication with the system control logic 108.

In some embodiments, the system control logic 108 may include one or more memory controllers to provide an interface connected to the system memory 112. The system memory 112 may be used to load and store data and/or instructions. In some embodiments, the memory 112 of the electronic device 100 may include any suitable volatile memory, such as a suitable dynamic random access memory (DRAM). In some embodiments, the system memory 112 may be used to load or store instructions for implementing the above image processing, or the system memory 112 may be used to load or store instructions for implementing an application program to perform image processing by using the neural network model.

The NVM/memory 116 may include one or more tangible, non-transitory

computer-readable media for storing data and/or instructions. In some embodiments, the NVM/memory 116 may include any suitable nonvolatile memory such as a flash memory and/or any suitable nonvolatile storage device, such as at least one of a hard disk drive (Hard Disk Drive, HDD), a compact disc (Compact Disc, CD) drive, and a digital versatile disc (Digital Versatile Disc, DVD) drive. The NVM/memory 116 can be further configured to store the neural network model used in the above image processing.

The NVM/memory 116 may include a part of storage resources on an apparatus on which the electronic device 100 is installed, or may be accessed by the device, but is not necessarily a part of the device. For example, the NVM/storage 116 can be accessed by using the network via the network interface 120.

In particular, the system memory 112 and the NVM/memory 116 may respectively include: a temporary copy and a permanent copy of an instruction 124. The instruction 124 may include an instruction that the electronic device 100 is caused to implement the image processing method shown in this application when the instruction is executed by at least one of the processors 104. In some embodiments, the instruction 124, and hardware, firmware, and/or software components thereof may additionally/alternatively be placed in the system control logic 108, the network interface 120, and/or the processor 104.

The network interface 120 may include a transceiver for providing a radio interface for the electronic device 100, to further communicate with any other suitable devices (such as a front-end module or an antenna) by using one or more networks. In some embodiments, the network interface 120 may be integrated with other components of the electronic device 100. For example, the network interface 120 may be integrated into at least one of the processor 104, the system memory 112, the NVM/memory 116, and a firmware device (not shown) with an instruction, and when at least one of the processors 104 executes the instruction, the electronic device 100 implements the image processing method shown in this application.

The network interface 120 may further include any suitable hardware and/or firmware to provide a multiple-input multiple-output radio interface. For example, the network interface 120 may be a network adapter, a wireless network adapter, a telephone modem, and/or a wireless modem.

In some embodiments, at least one of the processors 104 may be packaged with logic for one or more controllers of the system control logic 108 to form a system in package (SiP). In

some embodiments, at least one of the processors 104 may be integrated on the same die with the logic for one or more controllers of the system control logic 108 to form a system on chip (SoC).

The electronic device 100 may further include: an input/output (I/O) device 132. The I/O device 132 may include a user interface, so that a user can interact with the electronic device 100. A peripheral component can also interact with the electronic device 100 by using the design of a peripheral component interface. In some embodiments, the electronic device 100 further includes a sensor for determining at least one of environmental conditions and location information related to the electronic device 100.

According to some embodiments of the present application, an image processing method is provided. Figure 4 is a flowchart of an image processing method according some embodiments of the present application. As shown in figure 4, the method for performing image processing by using a MobileNeXt model includes the following steps:

Step 402: Obtain an image.

Step 404: The image is processed into at least one layer of image data on an input layer of the MobileNeXt model.

Step 406: The image data is processed by the first convolution layer including 32 output channels to obtain 32 output channels of feature maps of the image data.

Step 408: After the process of a fully convolutional layer, the local high-dimensional feature data of the image data is obtained. In the MobileNeXt model, the expansion ratio is set to be 6 by default;

Step 410: Being processed by the global average pooling layer, the two-dimensional feature map is converted into a one-dimensional feature vector, and the local high-dimensional feature data of the image data is pooled as the feature map of the image;

Step 412: The feature map is outputted by the process of the output layer.

In some embodiments, the hidden layer of the MobileNeXt model includes multiple processing units (also called sandglass blocks in the present application), and at least some of the multiple sandglass blocks are interconnected with each other and connected to the output of the input layer.

In some embodiments, in at least one sandglass block, the high-dimensional feature data of

the image as the input of the at least one sandglass block is undergone the shortcut path process which sequentially includes the first depthwise separable convolution layer, the first pointwise convolution layer, and the second pointwise convolution, and the second depthwise convolution layer, and the local high-dimensional feature data of the image is outputted.

In other embodiments, a fully connected layer may be added at the end of process, to predict the final score of each category.

Compared with the image processing method in the prior art, In the process of image processing the present application is capable of:

- (1) alleviate information loss and gradient confusion effectively.
- (2) reduce computational costs and remove of redundant parameters from the neural network effectively.
- (3) reduce computational complexity and improve classification accuracy of images.
- (4) the internal structure of the computer system can be more streamlined, allowing image processing to be applied in lightweight neural networks such as mobile devices.

Next, with reference to the drawings, the improvements made by the sandglass block in the present application with respect to the classic replacement structure and the inverted residual block will be described in detail.

According to some embodiments of the present application, figure 5(a) is a network structural diagram of a bottleneck structure in which standard convolution is replaced by depthwise separable convolution. As shown in figure 5(a), the first variant is constructed by directly modify the classic bottleneck structure, and the method is replacing the standard 3×3 convolution with a 3×3 depthwise convolution. In Table 2, there is listed several different structures from the performance comparison table. As shown in Table 2, compared with the sandglass block in this application, the performance of the first variant is reduced by about 5%. The main reason is that the depthwise convolution is performed in a bottleneck with a low-dimensional feature space, so that enough spatial information cannot be captured.

According to some embodiments of the present application, figure 5(b) is a network structure diagram of a bottleneck structure in which the standard convolution is replaced by two depthwise separable convolutions. As shown in figure 5(b), the second variant has been improved on the basis of the first variant. The improvement is that another 3×3 depthwise convolution is added to the bottleneck. It can be learned that, compared with the structure shown in figure 5(a), the top-1 accuracy of the second variant is improved by more than 1%. It shows that encoding more spatial information can indeed bring good results. The same result can also be obtained by comparing figure 5(b) and figure 5(d).

Figure 5(c) is a network structural diagram of two pointwise convolution positions in a switching sandglass block according to some embodiments of the present application. As shown in figure 5(c), the third variant moves the depthwise separable convolution from the high-dimensional feature space to the bottleneck position with less feature channels based on the original inverted residual block. Compared with figure 5(b), this variant has more learnable parameters and calculation costs, but the performance is worse. This also shows that creating shortcuts between high-dimensional representations is more beneficial to network performance.

Figure 5(d) is a network structural diagram of a sandglass block according to some embodiments of the present application. The sandglass block in the present application can obtain better performance than the other three variants. Comparison of performance is shown in Table 2.

Table2:

Block structure	#Dwise conv	Param.(M)	M-Adds. (M)	Top-1 Acc. (%)
MobileNetV2	1	3.5	300	72.30
Figure 5(a)	1	3.4	240	68.90
Figure 5(b)	2	3.4	250	70.11

Figure 5(c)	2	3.5	300	69.26
Figure 5(d)	2	3.5	300	74.02

Specifically, the reason for the performance improvement can be explained by the following two rules:

(1) To preserve more information from the bottom layers and to facilitate the gradients propagation across layers when transiting to the top layers, the shortcuts should be used to connect high-dimensional representations;

(2) The depthwise separable convolutions with a small kernel size (e.g., 3×3) are light-weight, and therefore a couple of depthwise separable convolutions onto the high-dimensional features can be appropriately applied such that richer spatial information can be encoded to generate more expressive representations.

Table 3 shows the network structure of the sandglass block. The sandglass block in the present application proves that the method of establishing a shortcut between high-dimensional representations in the classic bottleneck structure is also applicable to lightweight neural networks. It is understood that this is the first time to use the advantages of the classic bottleneck structure over the inverted residual block to conduct a research on an effective neural network design. It also proves that, by adding depthwise separable convolution at the end of the residual path of the sandglass block in the present application, the neural network can be encouraged to learn more expressive spatial information, resulting in better performance.

Table 3:

Input dimension	Operator type	Output dimension
$D_f \times D_f \times M$	3×3 Dwise conv, ReLU6	$D_f \times D_f \times M$
$D_f \times D_f \times M$	1×1 conv, linear	$D_f \times D_f \times \frac{M}{t}$

$D_f \times D_f \times \frac{M}{t}$	1×1 conv,ReLU6	$D_f \times D_f \times N$
$D_f \times D_f \times N$	3×3 Dwise conv, linear, stride=s	$\frac{D_f}{s} \times \frac{D_f}{s} \times N$

In some embodiments, as shown in figure 5(d), the first pointwise convolution layer performs reduction on the image data, and the second pointwise convolution layer performs expansion on the image data outputted by the first pointwise convolution layer. As shown in figure 5(c), in the inverted residual block, expansion is performed at first and then reduction is performed. In order to ensure the shortcut connection to high-dimensional representations, the sandglass block in the present application proposes to reverse the order of the two pointwise convolutions. $F \in \mathbb{R}^{D_f \times D_f \times M}$ is set to be the input tensor of the sandglass block and $G \in \mathbb{R}^{D_f \times D_f \times M}$ is set to be the output tensor. The depthwise separable convolution and activation layers are not considered at this moment. Then the formula of the sandglass block can be written as follows:

$$G = \Phi_e(\Phi_r(F)) + F, \quad (1)$$

Where Φ_e and Φ_r denote the pointwise convolutions for channel expansion and reduction, respectively. In this way, the bottleneck can be kept in the middle of the residual path for reducing parameters and calculation costs. Furthermore, this allows us to use the shortcut connection to connect representations with a large number of channels instead of the bottleneck ones.

In some embodiments, no other convolution operations are performed between the first pointwise convolutional layer and the second pointwise convolutional layer. Pointwise convolutions can be used to encode inter-channel information, but cannot capture spatial information. In the sandglass block of the present application, depthwise separable convolution is used to encode spatial information. The inverted residual block adds depthwise separable convolutions between point-wise convolutions to learn expressive spatial context information.

figure 6 is a diagram of comparison between a classic bottleneck structure with depthwise separable convolution and a network structure of a sandglass block according to some embodiments of the present application. As shown in figure 6, in the sandglass block of the present application, there is a bottleneck structure between two pointwise convolutions. The depthwise separable convolution is not directly put between the two pointwise convolutions, but is added at the ends of the residual path, while no other convolution operations are performed between the two pointwise convolutions. Mathematically, the basic components can be formulated as follow:

$$\hat{G} = \Phi_{1,p} \Phi_{1,d}(F), \quad (2)$$

$$G = \Phi_{2,d} \Phi_{2,p}(\hat{G}) + F, \quad (3)$$

Where $\Phi_{i,p}$ and $\Phi_{i,d}$ denote the i th pointwise convolution and depthwise convolution, respectively. Because two depthwise convolutions are conducted in a high-dimensional space, richer feature representations can be extracted as compared with the inverted residual block. Adding depthwise separable convolution directly to the bottleneck can reduce the number of filters, so that less spatial information needs to be encoded. By experiments, it is found that compared with MobileNetV2, the structure greatly reduces the performance by more than 1%.

In some embodiments, the shortcut path is established in the high-dimensional feature space of the image data, and the local feature tensor is extracted to obtain the local high-dimensional feature data of the image data. Instead of putting the shortcut between bottlenecks, the sandglass block puts the shortcuts between higher-dimensional representations. As shown in figure 6, compared with the inverted residual block, the "wider" shortcut connection than the bottleneck structure can deliver more information from the input tensor F to the output tensor G and allows more gradients to propagate across multiple layers. The shortcut connection is established only when the input channel is consistent with the output channel.

The shortcut connection can improve the cross-layer propagation gradient efficiency in the residual block. Experiments show that it is not necessary to extract the global identity tensor of

the residual block, but only the local feature tensor of the image. In order to make the network more adaptable to mobile devices, a new hyper-parameter identification—identity tensor multiplier that is denoted as $\alpha \in [0,1]$ is introduced, and is used to determine what portion of the channels in the identity tensor is preserve, that is, a local feature tensor of a corresponding portion of the image is extracted. For convenience, Φ is set to be the transformation function of the residual path in the sandglass block. Initially, the formula of the sandglass block can be written as $G=\Phi(F)+F$. After applying the multiplier, the sandglass block can be rewritten as

$$G_{1:\alpha M} = \Phi(F)_{1:\alpha M} + F_{1:\alpha M}, G_{\alpha:\alpha M} = \Phi(F)_{\alpha:\alpha M}, \quad (4)$$

Where the subscript represents the dimension of the channel.

The advantages of using α are shown in two aspects. In a first aspect, after reducing the multiplier, the number of element-wise additions in each sandglass block can be reduced. As we all know, element-wise addition is time-consuming. Choosing a local identity tensor multiplier can result in better delays with little performance degradation. In a second aspect, the number of memory accesses can be reduced. One of the main factors that affect the model latency is the memory access cost (MAC). Because the shortcut connection identity tensor is from the output of the last building block, its recurrent nature hints an opportunity to cache it on the chip in order to avoid the excessive off-chip memory access. Therefore, reducing the channel dimension of the identity tensor can effectively encourage the processors to store it in the cache or other faster memory near the processors and hence improve the latency.

In some embodiments, a linear activation function is used for the data outputted by the first depthwise separable convolutional layer; a nonlinear activation function is used for the data outputted by the first pointwise convolutional layer; a linear activation function is used for the data outputted by the second pointwise convolutional layer; a nonlinear activation function is used for the data outputted by the second depthwise convolutional layer. It should be clear to the person skilled in the art that the use of linear bottlenecks can prevent the eigenvalues from returning to zero, thereby reducing information loss. Therefore, any activation layers after the

reduction layer (the first pointwise convolution layer) are not added. It should be also clear to the person skilled in the art that, though the output of the building block is high-dimensional features, adding an activation layer after the last convolution can negatively influence the classification performance. Therefore, activation layers are only added after the first depthwise convolutional layer and the last pointwise convolutional layer.

According to some embodiments of the present application, an image processing apparatus 600 is provided. Figure 7 illustrates a structural schematic of an image processing apparatus in accordance with some embodiments of the present application. As shown in figure 7, the apparatus 600 for video motion recognition using the MobileNeXt model is as follows:

obtaining module 602, obtaining an image;

processing module 604, processing the image into at least one layer of image data based a neural network model; inputting the image data for processing to obtain local high-dimensional feature data of the image data, wherein the processing module comprises a plurality of processing units, such as processing unit1, processing unit2, processing unit3,etc. And at least one or more processing units among the plurality of processing units are interconnected and connected to the outputting module; and

outputting module 606, outputting a feature map of the image, wherein the feature map of the image is obtained by pooling and fusing local high-dimensional features of the image data; wherein

in at least one processing unit, the high-dimensional feature data of the image as input of the at least one processing unit is undergone a shortcut connection process which sequentially comprise a first depthwise separable convolutional layer, a first pointwise convolutional layer, a second pointwise convolutional layer, and a second depthwise separable convolutional layer, and the local high-dimensional feature data of the image is outputted.

In some embodiments, the apparatus further includes that the first pointwise convolutional layer performs channel reduction on the image data, and the second pointwise convolutional layer performs channel expansion on the image data outputted by the first pointwise convolutional layer. Not only parameters and calculation costs are reduced, but also a shortcut connection instead of a bottleneck connection can be used to connect a representation with a

large number of channels.

In some embodiments, the apparatus further includes that no other convolution operations are performed between the first pointwise convolutional layer and the second pointwise convolutional layer. This structure greatly reduces performance by more than 1%.

In some embodiments, the apparatus further includes that the shortcut connection is established in a high-dimensional feature space of the image data, and a local feature tensor of the image data is extracted, to obtain the local high-dimensional feature data of the image data.

In some embodiments, the apparatus further includes that a linear activation function is used for data outputted by the first depthwise separable convolutional layer; a nonlinear activation function is used for data outputted by the first pointwise convolutional layer; a linear activation function is used for data outputted by the second pointwise convolutional layer; and a nonlinear activation function is used for data outputted by the second depthwise convolutional layer. In this way, more information can be passed from an input tensor to an output tensor, and more gradients are allowed to propagate across multiple layers.

In some embodiments, the apparatus further includes that the neural network model includes an input layer, multiple hidden layers, and an output layer, the hidden layer further includes a first convolutional layer, a fully convolutional layer, and a global average pooling layer, the fully convolutional layer includes at least one processing unit, the input image data is processed by the first convolutional layer including 32 output channels to obtain 32 feature maps of the image data. The feature map of the image data is processed by the fully convolutional layer to obtain the local high-dimensional feature data of the image data. The local high-dimensional feature data of the image data is processed by the global average pooling layer and is pooled as the feature map of the image.

Next, advantages of MobileNeXt over MobileNetV2 are explained from different perspectives with reference to experimental data.

The experiment is based on a model trained by using a PyTorch toolbox and using a standard SGD optimizer, where both attenuation and momentum of the model are 0.9 and weight attenuation is 4×10^{-5} , a cosine learning schedule has an initial learning rate of 0.05, a batch size is set to be 256, and four GPUs are used for training. In addition to comparing the performance of the complete model (that is, the weight multiplier is 1) for classification, performance is compared with other weight multipliers and quantization, and performance advantages of the

MobileNeXt model in terms of overall model architecture configuration are reflected.

Comparisons with different width multipliers are performed by using five different width multipliers of 1.4, 1.0, 0.75, 0.5, and 0.35 as scale factors and based on different width multipliers with an input resolution of 224×224 , to measure complexity and accuracy of the used model. As shown in Table 4, when testing on Google Pixel 4XL with PyTorch environment settings (for example, two models with a width multiplier of 1.0 is 210ms) is performed. The smaller the multiplier is set to be, the better performance can be obtained through MobileNetV2. The MobileNeXt model with different multipliers is superior to MobileNetV2 in terms of learnable parameters and calculation costs. When the multiplier is smaller, the performance gain of the MobileNeXt model is particularly higher compared with MobileNetV2.

Table 4

No.	Method	Param. (M)	M Adds (M)	Top-1 Acc. (%)
1	MobileNetV2-1.40	6.9	690	74.9
2	MobileNetV2-1.00	3.5	300	72.3
3	MobileNetV2-0.75	2.6	150	69.9
4	MobileNetV2-0.50	2.0	97	65.4
5	MobileNetV2-0.35	1.7	59	60.3
6	MobileNeXt-1.40	6.1	590	76.1
7	MobileNeXt-1.00	3.4	300	74.0
8	MobileNeXt-0.75	2.5	210	72.0
9	MobileNeXt-0.50	2.1	110	67.7
10	MobileNeXt-0.35	1.8	80	64.7

Through experimental verification, in the following cases, MobileNeXt can be combined with a quantization method to achieve better performance than MobileNetV2. As shown in Table 5, in the bit configuration, "W" represents the number of model weights, and "A" represents the

number of activated bits. When quantization is not performed, the MobileNeXt model improves top-1 accuracy by more than 1.7% relative to MobileNetV2. Currently, a more common general solution is 8-bit quantization, and therefore the parameters and activation quantization are set to be 8 bits. In this case, under the same quantization setting, performance of the MobileNeXt model is 3.55% higher than that of MobileNetV2. There are two reasons for such a big improvement: First, compared with MobileNetV2, the shortcut connection in each processing unit is converted from a low-dimensional representation to a high-dimensional representation, so that more useful feature representations can be retained after quantization. Second, more spatial information can be retained by using more depth separable convolutions, and this helps improve classification performance.

Table 5

Model	Precision (W/A)	Method	Top-1 Acc. (%)
MobileNetV2	INT8/INT8	Post Training Quant.	65.07
MobileNeXt	INT8/INT8	Post Training Quant.	68.62
MobileNetV2	FP32/FP32	-	72.25
MobileNeXt	FP32/FP32	-	74.02

As shown in figure 6, the processing unit in the present application includes two 3×3 depth separable convolutions, which are used to encode rich spatial context information. In order to prove that it is the processing unit that makes the MobileNeXt model more optimized, in this experiment, a structural comparison with an improved version of MobileNetV2 is performed, and the improvement consists in inserting a deep convolution in the middle of each inverted residual block.

Table 6

Method	#Depthwise convolution	Param. (M)	M-Adds(M)	Top-1 Acc. (%)
MobileNetV2	2 (Middle)	3.6	340	73.02

MobileNeXt	2 (Top, bottom)	3.5	300	74.02
------------	-----------------	-----	-----	-------

The results are shown in Table 6. The performance of the improved version of MobileNetV2 has increased to 73%, but there is still a certain gap compared with the performance of the MobileNeXt model that is 74%, and meanwhile, this leads to more parameters and higher complexity. This shows that in terms of network structure, the MobileNeXt model has more advantages than MobileNetV2.

In order to further verify superiority of the processing unit in the present application relative to the inverted residual block, a SE block (Squeeze-and-Excitation) is added to the MobileNeXt model, an EfficientNet-b0 architecture is directly used, and a basic expansion strategy replaces the inverted residual block with the processing unit. As shown in Table 7, MobileNeXt represents a model with only processing units, and MobileNeXt+ represents a model with added processing units and SE blocks. In order to make a fair comparison with other latest models, no search method is applied to the architectural design and data augmentation. A considerable amount of calculation and a parameter reduction ratio of 20% are used, and when the processing unit is used to replace the inverted residual block, the top-1 classification accuracy of the ImageNet-1k dataset can be improved by 0.4%.

Table 7

Method	Param. (M)	MAdd(M)	Top-1 Acc. (%)
MobilenetV1-1.0	4.2	575	70.6
ShuffleNetV2-1.5	3.5	299	72.6
MobilenetV2-1.0	3.5	300	72.3
MnasNet-A1	3.9	312	75.2
MobilenetV3-L-0.75	4.0	155	73.3
ProxylessNAS	4.1	320	74.6

FBNet-B	4.5	295	74.1
IGCV3-D	7.2	610	74.6
GhostNet-1.3	7.3	226	75.7
EfficientNet-b0	5.3	390	76.3
MobileNeXt-1.0	3.4	300	74.02
MobileNeXt-1.0+	3.98	330	76.05
MobileNeXt-1.1+	4.28	420	76.7

According to some embodiments of the present application, an application for applying the model to an object detection task is provided.

The MobileNeXt model and MobileNetV2 are separately used as feature extractors, and a Pascal VOC 2007 test set is used for testing based on an SSDLite detector. The first layer and the second layer of SSDLite are set to be connected to the last pointwise convolutional layer by using output strides of 16 and 32, respectively. The remaining SSDLite layers are connected to the top of the last pointwise convolutional layer by using an output stride of 32. During training, a batch size of 24 is used, and all models are trained for 240,000 iterations. Table 8 shows training results obtained when different training models are used. As shown in Table 8, when the same SSDLite320 detector is used, and better results on the Mean Average Precision (mAP, Mean Average Precision) can be obtained by replacing the MobileNetV2 model with MobileNeXt. Obviously, with almost the same number of parameters and the same amount of calculation, SSDLite with the MobileNeXt model improves parameters and performance of MobileNetV2 by nearly 1%. This shows that the MobileNeXt model has better transferability compared with MobileNetV2.

Table 8

No.	Method	Training model	Param. (M)	M-Adds. (B)	mAP (%)
1	SSD300	VGG	36.1	35.2	77.2

2	SSDLite320	MobileNetV2	4.3	0.8	71.7
3	SSDLite320	MobileNeXt	4.3	0.8	72.6

According to some embodiments of the present application, an application that treats the processing unit as a "super" operator with a more powerful conversion capability than a conventional convolution operator is provided.

In some embodiments, a calculation unit for searching is provided and used as a basic calculation unit for finally performing structure searching. In order to further study superiority of the processing unit in the present application relative to the inverted residual block, the processing unit and the inverted residual block are separately added to the search space of the Differentiable Architecture Search (DARTS) to view the network performance after the architecture search, and view the corresponding results on the dataset CIFAR-10.

As mentioned above, the DARTS search algorithm is used to search for a cell structure. In this embodiment, the search space including the following basic operations is used as the baseline (denoted as the original DARTS):

- 1) Convolution operation (ConvOp): regular convolution (reg_conv), dilation convolution (dil_conv), separable convolution (sep_conv), and depthwise convolution (dpwise_conv);
- 2) Convolution kernel size (kernel size): 3×3 , 5×5 , 7×1 followed by 1×7 ;
- 3) Non-parametric operations: average pooling (ave_pool), maximum pooling (max_pool), skip connection (skip connection), and none (None).

In order to be compared with the inverted residual block, architecture search is performed in the following two new search spaces in this embodiment:

- 1) DARTS + an inverted residual block: The original search space plus the inverted residual block is used as a candidate for depthwise separable convolution operations.
- 2) DARTS + a processing unit according to this application: The original search space plus the processing unit according to this application.

In the search process, the training strategy is strictly followed, the same hyperparameters

are used for fair comparison, and the number of nodes is set to be 7 in each directed acyclic graph (DAG) of the cell. Figure 8(a) is a schematic diagram of a process of searching for a normal cell structure on a CIFAR-10 dataset by using DARTS according to some embodiments of the present application. Figure 8(b) is a schematic diagram of a process of searching for a reduced cell structure on a CIFAR-10 dataset by using DARTS according to some embodiments of the present application. "SGBlock" represents the processing unit in the present application. In the search space, as shown in Figure 8(a), the processing unit is used to search for the normal cell structure, separable convolution operations and skip connection operations are replaced, and the processing unit is used as the only operation unit. As shown in Figure 8(b), when searching for the reduced cell is performed, no operation of the processing unit is required.

Table 9

No.	Search space	Test error rate (%)	Param. (M)	Search method	#Operators
1	Original DARTS	3.11	3.25	based on gradient	7
2	DARTS+Inverted residual block	3.26	3.29	based on gradient	7
3	DARTS+Processing unit	2.98	2.45	based on gradient	7

The results are shown in Table 9. The processing unit used as a new operator is added to the DARTS search space under a normal cell structure, so that the obtained model has higher accuracy than a model with only the original DARTS search space, and the parameter is reduced by about 25%. However, the search model with the inverted residual block added to the search space reduces performance of the original model. Adding the processing unit to the original search space can greatly reduce the number of learnable parameters in the search architecture and improve classification performance of the CIFAR-10 dataset. It is shown again that, when a search algorithm is combined for usage, the processing unit in this application can be used to

replace the original block to improve performance.

Figure 9 is a schematic structural diagram of an electronic device for image processing according to some embodiments of the present application. As shown in Figure 9, an electronic device 800 includes:

- an image obtaining module 802, configured to obtain an image;
- an input layer 804, processing the image into at least one layer of image data;
- a first convolution layer 806, processing the image data into a feature map;
- a fully convolutional layer 808, outputting the feature map as local high-dimensional feature data of the image;
- a global average pooling layer 810, pooling the local high-dimensional feature data of the image data as a feature map of the image; and
- an output layer 812, outputting the feature map of the image.

It can be understood that the electronic device 800 for image processing shown in Figure 9 corresponds to the image processing method provided by the present application, and the technical details in the above specific description about the image processing method provided by the present application are still applicable to the electronic device 800 for image processing shown in Figure 9. For a specific description, refer to the above description. Details are repeated herein.

It can be understood that exemplary applications of neural network-based image processing provided by the embodiments of the present application include but are not limited to image processing in the field of artificial intelligence.

Figure 10 is a schematic structural diagram of an electronic device 900 according to an embodiment of the present application. The electronic device 900 can also execute the image processing method disclosed in the above embodiments of the present application. In Figure 10, similar components have the same reference signs. As shown in Figure 10, the electronic device 900 may include a processor 910, a power module 940, a memory 980, a mobile communication module 930, a wireless communication module 920, a sensor module 990, an audio module 950, a camera 970, an interface module 960, a key 901, a display screen 902, and so on.

It can be understood that the structure illustrated in this embodiment of the present application does not constitute a specific limitation on the electronic device 900. In other embodiments of the present application, the electronic device 900 may include more or less components than shown, or some components are combined, or split some components are split, or different components are arranged. The illustrated components can be implemented in hardware, software, or a combination of software and hardware.

The processor 910 may include one or more processing units, for example, may include a processing unit or a processing circuit such as a central processing unit (Central Processing Unit, CPU), a graphics processing unit (Graphics Processing Unit, GPU), a digital signal processor (Digital Signal Processor, DSP), a micro-programmed control unit (Micro-Programmed Control Unit, MCU), an artificial intelligence (Artificial Intelligence, AI) processor, or a field programmable gate array (Field Programmable Gate Array, FPGA). Different processing units may be independent devices, or may be integrated in one or more processors. A storage unit is disposed in the processor 910, and is configured to store instructions and data. In some embodiments, the storage unit in processor 910 is a cache memory 980. The memory 980 mainly includes a storage program area 9801 and a data storage area 9802. The storage program area 9801 may store an operating system and application programs required by at least one function (functions such as voice playback or image recognition). The image processing method based on the neural network model provided in the embodiment of the present application can be regarded as an application program in the stored program area 9801 to implement functions such as image processing. The neural network model is stored in the above data storage area 9802.

The power module 940 may include a power supply, a power management component, and the like. The power supply may be a battery. The power management component is used to manage charging of the power supply and supplying power to other modules by the power supply. In some embodiments, the power management component includes a charging management module and a power management module. The charging management module is configured to receive charging input from a charger. The power management module is configured to connect the power supply, the charging management module, and the processor 910. The power management module receives input from the power supply and/or the charging management module, and supplies power to the processor 910, the display screen 902, the camera 970, and the wireless communication module 920.

The mobile communication module 930 may include, but is not limited to, an antenna, a power amplifier, a filter, a low noise amplifier (Low Noise Amplifier, LNA), and the like. The mobile communication module 930 may provide a solution of wireless communication including 2G/3G/4G/5G and the like applied to the electronic device 900. The mobile communication module 930 can receive electromagnetic waves from the antenna, filter and amplify the received electromagnetic waves, and transmit the electromagnetic waves to a modem processor for demodulation. The mobile communication module 930 can further amplify the signal modulated by the modem processor, and convert the signal into electromagnetic waves for radiation through the antenna. In some embodiments, at least some of the functional modules of the mobile communication module 930 may be disposed in the processor 910. In some embodiments, at least some of the functional modules of the mobile communication module 930 may be arranged in the same device as at least some of the modules of the processor 910. The wireless communication technology may include global system for mobile communications (global system for mobile communications, GSM), general packet radio service (general packet radio service, GPRS), Code Division Multiple Access (code division multiple access, CDMA), and Wideband Code Division Multiple Access (wideband code division multiple access, WCDMA), Time Division-Synchronous Code Division Multiple Access (Time Division-Synchronous Code Division Multiple Access, TD-SCDMA), Long Term Evolution (Long Term Evolution, LTE), Bluetooth (Bluetooth, BT), global navigation satellite system (global navigation satellite system, GNSS), wireless local area network (wireless local area network, WLAN), near field communication (near field communication, NFC), frequency modulation (frequency modulation, FM), an infrared (infrared, IR) technology, etc. The GNSS may include a global positioning system (global positioning system, GPS), a Global Navigation Satellite System (Global Navigation Satellite System, GLONASS), a BeiDou Navigation Satellite System (BeiDou Navigation Satellite System, BDS), a Quasi-Zenith Satellite System (Quasi-Zenith Satellite System, QZSS), and/or a Satellite Based Augmentation System (Satellite Based Augmentation System, SBAS).

The wireless communication module 920 may include an antenna, and transmitting and receiving of electromagnetic waves are implemented via the antenna. The wireless communication module 920 can provide a solution of wireless communication including wireless local area network (wireless local area network, WLAN) (such as wireless fidelity (wireless fidelity, Wi-Fi) network), Bluetooth (Bluetooth, BT), and global navigation satellite system (global navigation satellite system, GNSS), frequency modulation (FM), a near field

communication (near field communication, NFC) technology, or an infrared (infrared, IR) technology that is applied to the electronic device 900. The electronic device 900 can communicate with the network and other devices by using the wireless communication technology.

In some embodiments, the mobile communication module 930 and the wireless communication module 920 of the electronic device 900 may be located in the same module.

The display screen 902 is used to display human-computer interaction interfaces, images, videos, and the like. The display screen 902 includes a display panel. The display panel may use a liquid crystal display (liquid crystal display, LCD), an organic light-emitting diode (organic light-emitting diode, OLED), or an active matrix organic light-emitting diode (active-matrix organic light emitting diode, AMOLED), a flexible light-emitting diode (flexible light-emitting diode, FLED), Miniled, MicroLed, Micro-oled, a quantum dot light emitting diode (quantum dot light emitting diode, QLED), etc.

The sensor module 990 may include a proximity light sensor, a pressure sensor, a gyro sensor, an air pressure sensor, a magnetic sensor, an acceleration sensor, a distance sensor, a fingerprint sensor, a temperature sensor, a touch sensor, an ambient light sensor, a bone conduction sensor, and the like.

The audio module 950 is configured to convert digital audio information into an analog audio signal for outputting, or convert analog audio input into a digital audio signal. The audio module 950 may further be configured to encode and decode audio signals. In some embodiments, the audio module 950 may be disposed in the processor 910, or some of the functional modules of the audio module 950 may be disposed in the processor 910. In some embodiments, the audio module 950 may include a loudspeaker, an earpiece, a microphone, and a headset jack.

The camera 970 is used to capture static images or videos. The object generates an optical image by using a lens and projects the optical image onto a photosensitive element. The photosensitive element converts an optical signal into an electrical signal, and then transmits the electrical signal to an image signal processor (Image Signal Processor, ISP) to be converted into a digital image signal. The electronic device 900 can implement a shooting function by using an ISP, the camera 970, a video codec, a graphics processing unit (Graphic Processing Unit, GPU), the display screen 902, an application processor, and the like.

The interface module 960 includes an external memory interface, a universal serial bus (universal serial bus, USB) interface, and a subscriber identification module (subscriber identification module, SIM) card interface. The external memory interface may be used to connect an external memory card, such as a Micro SD card, to expand a storage capacity of the electronic device 900. The external memory card communicates with the processor 910 by using the external memory interface to implement a data storage function. The universal serial bus interface is used for the electronic device 900 to communicate with other electronic devices. The user identification module card interface is used to communicate with the SIM card installed in the electronic device 900, for example, to read a phone number stored in a SIM card or write the phone number into the SIM card.

In some embodiments, the electronic device 900 further includes a key 901, a motor, an indicator, and the like. The key 901 may include a volume key, an on/off key, and so on. The motor is used to cause the electronic device 900 to generate a vibration effect. For example, when the electronic device 900 of a user is called, vibration is generated to remind the user of answering an incoming call of the electronic device 900. The indicator may include a laser indicator, a radio frequency indicator, an LED indicator, and the like.

The embodiments of the mechanism disclosed in the present application may be implemented in hardware, software, firmware, or a combination of these implementation methods. The embodiments of the present application may be implemented as a computer program or program code executed on a programmable system. The programmable system includes at least one processor, a storage system (including a volatile and non-volatile memory and/or a storage element), at least one input device, and at least one output device.

The program code can be applied to input instructions to perform the functions described in the present application and generate output information. The output information can be applied to one or more output devices in a known manner. For the objectives of the present application, a processing system includes any system having a processor such as a digital signal processor (DSP), a microcontroller, an application specific integrated circuit (ASIC), or a microprocessor.

The program code can be implemented in a high-level programming language or an object-oriented programming language to communicate with the processing system. Alternatively, the program code can be implemented in assembly language or machine language if necessary. In fact, the mechanism described in the present application is not limited to the scope of any particular programming language. In either case, the language can be a compilation

language or an interpreted language.

In some cases, the disclosed embodiments may be implemented in hardware, firmware, software, or any combination thereof. The disclosed embodiments can further be implemented as instructions carried by or stored on one or more transitory or non-transitory machine-readable (for example, computer-readable) storage media, and the instructions can be read and executed by one or more processors. For example, the instructions may be distributed by using the network or by using other computer-readable media. Therefore, the machine-readable media may include any mechanism for storing or transmitting information in a machine-readable (for example, a computer-readable) form, including but not limited to, a floppy disk, a compact disk, an optical disc, a read-only memory (CD-ROM), a magneto optical disc, a read only memory (ROM), a random access memory (RAM), an erasable programmable read only memory (EPROM), an electrically erasable programmable read only memory (EEPROM), a magnetic card or an optical card, a flash memory, or a tangible machine-readable memory used to transmit information (for example, a carrier, an infrared signal, and a digital signal) in an electrical, optical, or acoustic form, or other forms of propagating signals by using the Internet. Therefore, the machine-readable media include any type of machine-readable media suitable for storing or transmitting electronic instructions or information in a machine-readable (for example, computer-readable) form.

In the drawings, some structural or method features may be shown in a specific arrangement and/or order. However, it should be understood that such a specific arrangement and/or ordering may not be needed. Rather, in some embodiments, these features may be arranged in a manner and/or order different from the manner and/or order shown in the illustrative drawings. In addition, the inclusion of structural or method features in a particular figure does not imply that such features are needed in all embodiments, and these features may not be included or may be combined with other features in some embodiments.

It should be noted that each unit/module mentioned in each device embodiment of the present application is a logical unit/module. Physically, a logical unit/module may be a physical

unit/module or a part of a physical unit/module, and can further be implemented by a combination of multiple physical units/modules. The physical implementation of these logical units/modules is not the most important, but the combination of the functions implemented by these logical units/modules is the key to solving the technical problem raised by the present application. In addition, in order to highlight the innovative part of the present application, the above device embodiments of the present application do not introduce units/modules that are not closely related to solving the technical problem raised by the present application, but this does not mean that the above device embodiments do not include other units/modules.

It should be noted that, in the examples and the specification of this patent, relational terms such as "first" and "second" are used only to distinguish one entity or operation from another entity or operation, and do not necessarily require or imply that there is any such actual relationship or order between these entities or operations. Moreover, the terms "comprise", "include" or any other variant thereof are intended to cover non-exclusive inclusion, so that a process, a method, an article, or a device that includes a series of elements includes not only those elements, but also other elements not explicitly listed, or elements that are inherent to this process, method, article, or device. In the absence of more limitations, the element defined by the sentence "include one" does not exclude that there are other identical elements in the process, method, article or device that includes the element.

Although the present application has been illustrated and described by referring to certain preferred embodiments of the present application, persons of ordinary skill in the art should understand that various changes can be made in form and detail without departing from the spirit and scope of the present application.

Claims

1. An image processing method based on a neural network, comprising:

obtaining an image;

processing the image into at least one layer of image data on an input layer of the neural network model;

inputting the image data into a hidden layer of the neural network model for processing, so as to obtain local high-dimensional feature data of the image data, wherein the hidden layer of the neural network model comprises a plurality of processing units, and at least one or more processing units among the plurality of processing units are interconnected and connected to output of the input layer; and

outputting a feature map of the image, wherein the feature map of the image is obtained by pooling and fusing local high-dimensional features of the image data; wherein

in at least one processing unit, the high-dimensional feature data of the image as input of the at least one processing unit is undergone a shortcut connection process which sequentially comprise a first depthwise separable convolutional layer, a first pointwise convolutional layer, a second pointwise convolutional layer, and a second depthwise separable convolutional layer, and the local high-dimensional feature data of the image is outputted.

2. The method according to claim 1, wherein the first pointwise convolutional layer performs channel reduction on the image data, and the second pointwise convolutional layer performs channel expansion on the image data outputted by the first pointwise convolutional layer.

3. The method according to claim 2, wherein no other convolution operations are performed between the first pointwise convolutional layer and the second pointwise convolutional layer.

4. The method according to claim 3, the shortcut connection is established in a high-dimensional feature space of the image data, and a local feature tensor of the image data is extracted, to obtain the local high-dimensional feature data of the image data.

5. The method according to claim 4, wherein a linear activation function is used for data outputted by the first depthwise separable convolutional layer; a nonlinear activation function is used for data outputted by the first pointwise convolutional layer; a linear activation function is used for data outputted by the second pointwise convolutional layer; and a nonlinear activation function is used for data outputted by the second depthwise convolutional layer.

6. The method according to claims 1-5, wherein the neural network model comprises an input layer, multiple hidden layers, and an output layer, the hidden layer further comprises a first convolutional layer, a fully convolutional layer, and a global average pooling layer, the fully convolutional layer comprises at least one processing unit, the input image data is processed by the first convolutional layer comprising 32 output channels to obtain 32 output channels of feature maps of the image data;

the feature map of the image data is processed by the fully convolutional layer to obtain the local high-dimensional feature data of the image data;

the local high-dimensional feature data of the image data is processed by the global average pooling layer and is pooled as the feature map of the image.

7. An image processing apparatus, comprising:

obtaining module, obtaining an image;

processing module, processing the image into at least one layer of image data based a neural network model; inputting the image data for processing to obtain local high-dimensional feature data of the image data, wherein the processing module comprises a plurality of processing units, and at least one or more processing units among the plurality of processing units are interconnected and connected to the outputting module;; and

outputting module, outputting a feature map of the image, wherein the feature map of the image is obtained by pooling and fusing local high-dimensional features of the image data; wherein

in at least one processing unit, the high-dimensional feature data of the image as input of the at least one processing unit is undergone a shortcut connection process which sequentially comprise a first depthwise separable convolutional layer, a first pointwise convolutional layer, a second pointwise convolutional layer, and a second depthwise separable convolutional layer, and the local high-dimensional feature data of the image is outputted.

8. The apparatus according to claim 7, wherein the first pointwise convolutional layer performs channel reduction on the image data, and the second pointwise convolutional layer performs channel expansion on the image data outputted by the first pointwise convolutional layer.

9. The apparatus according to claim 8, wherein no other convolution operations are performed between the first pointwise convolutional layer and the second pointwise convolutional layer.

10. The apparatus according to claim 9, the shortcut connection is established in a high-dimensional feature space of the image data, and a local feature tensor of the image data is extracted, to obtain the local high-dimensional feature data of the image data.

11. The apparatus according to claim 10, wherein a linear activation function is used for data outputted by the first depthwise separable convolutional layer; a nonlinear activation function is used for data outputted by the first pointwise convolutional layer; a linear activation function is used for data outputted by the second pointwise convolutional layer; and a nonlinear activation function is used for data outputted by the second depthwise convolutional layer.

12. The apparatus according to claims 7-11, wherein the neural network model comprises an input layer, multiple hidden layers, and an output layer, the hidden layer further comprises a first convolutional layer, a fully convolutional layer, and a global average pooling layer, the fully convolutional layer comprises at least one processing unit, the input image data is processed by the first convolutional layer comprising 32 output channels to obtain 32 output channels of feature maps of the image data;

the feature map of the image data is processed by the fully convolutional layer to obtain the local high-dimensional feature data of the image data;

the local high-dimensional feature data of the image data is processed by the global average pooling layer and is pooled as the feature map of the image.

13. An image processing device, comprising:

a memory, configured to store instructions executed by one or more processors of a system, and

a processor, being one of the processors of the system, and configured to execute the instructions to implement the image processing method, wherein:

obtaining an image;

processing the image into at least one layer of image data on an input layer of the neural network model;

inputting the image data into a hidden layer of the neural network model for processing, so as to obtain local high-dimensional feature data of the image data, wherein the hidden layer of the neural network model comprises a plurality of processing units, and at least one or more processing units among the plurality of processing units are interconnected and connected to output of the input layer; and

outputting a feature map of the image, wherein the feature map of the image is obtained by pooling and fusing local high-dimensional features of the image data; wherein

in at least one processing unit, the high-dimensional feature data of the image as input of the at least one processing unit is undergone a shortcut connection process which sequentially

comprise a first depthwise separable convolutional layer, a first pointwise convolutional layer, a second pointwise convolutional layer, and a second depthwise separable convolutional layer, and the local high-dimensional feature data of the image is outputted.

14. The device according to claim 13, wherein the first pointwise convolutional layer performs channel reduction on the image data, and the second pointwise convolutional layer performs channel expansion on the image data outputted by the first pointwise convolutional layer.

15. The device according to claim 14, wherein no other convolution operations are performed between the first pointwise convolutional layer and the second pointwise convolutional layer.

16. The device according to claim 15, the shortcut connection is established in a high-dimensional feature space of the image data, and a local feature tensor of the image data is extracted, to obtain the local high-dimensional feature data of the image data.

17. The device according to claim 16, wherein a linear activation function is used for data outputted by the first depthwise separable convolutional layer; a nonlinear activation function is used for data outputted by the first pointwise convolutional layer; a linear activation function is used for data outputted by the second pointwise convolutional layer; and a nonlinear activation function is used for data outputted by the second depthwise convolutional layer.

18. The method according to claims 13-17, wherein the neural network model comprises an input layer, multiple hidden layers, and an output layer, the hidden layer further comprises a first convolutional layer, a fully convolutional layer, and a global average pooling layer, the fully convolutional layer comprises at least one processing unit, the input image data is processed by the first convolutional layer comprising 32 output channels to obtain 32 output channels of feature maps of the image data;

the feature map of the image data is processed by the fully convolutional layer to obtain the local high-dimensional feature data of the image data;

the local high-dimensional feature data of the image data is processed by the global average pooling layer and is pooled as the feature map of the image.

19. A computer-readable storage medium encoded with a computer program, wherein instructions are stored on the computer-readable medium, and when the instructions are executed on a computer, the computer performs the image processing method according to any one of claims 1-6.