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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
07 October 2021 (07.10.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/196070 A1

(51) International Patent Classification:
G06T 3/40 (2006.01) *G06N 3/02* (2006.01)
G06T 5/00 (2006.01)

(21) International Application Number:
PCT/CN2020/082722

(22) International Filing Date:
01 April 2020 (01.04.2020)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **BOE TECHNOLOGY GROUP CO., LTD.** [CN/CN]; No. 10 Jiuxianqiao Rd., Chaoyang District, Beijing 100015 (CN).

(72) Inventors: **CHEN, Guannan**; No. 9 Dize Rd., BDA, Beijing 100176 (CN). **WANG, Jingru**; No. 9 Dize Rd., BDA, Beijing 100176 (CN). **ZHANG, Lijie**; No. 9 Dize Rd., BDA, Beijing 100176 (CN). **HU, Fengshuo**; No. 9 Dize Rd., BDA, Beijing 100176 (CN).

(74) Agent: **TEE&HOWE INTELLECTUAL PROPERTY ATTORNEYS**; Suite 1, 6-12, 5th Floor, Tower W1, The Tower Offices, Oriental Plaza, No. 1 East Chang'an Avenue, Dongcheng District, Beijing 100738 (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK,

(54) Title: COMPUTER-IMPLEMENTED METHOD, APPARATUS, AND COMPUTER-PROGRAM PRODUCT

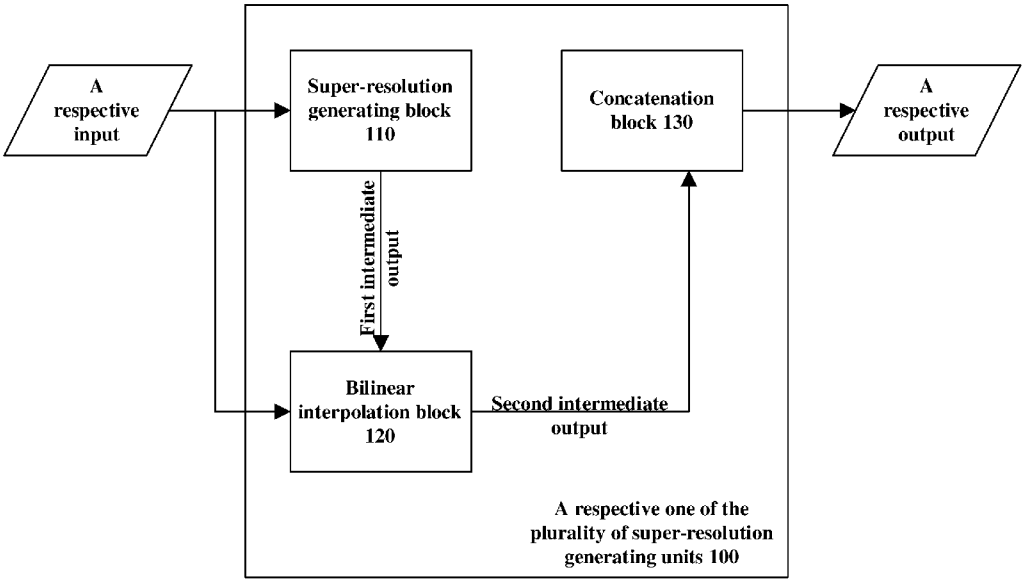


FIG. 4

(57) Abstract: A computer-implemented method is provided. The computer-implemented method includes inputting a low-resolution image into a generator; and generating a high-resolution image using the generator based on the low-resolution image. Generating the high-resolution image includes processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator. A respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units.

WO 2021/196070 A1

[Continued on next page]

EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

COMPUTER-IMPLEMENTED METHOD, APPARATUS, AND COMPUTER-PROGRAM PRODUCT

TECHNICAL FIELD

[0001] The present invention relates to display technology, more particularly, to a computer-implemented method, an apparatus, and a computer-program product.

BACKGROUND

[0002] Machine learning and neural networks have been used to analyze images for a variety of purposes. A neural network is a network including a plurality of hidden layers. A respective one of the plurality of hidden layers includes a plurality of neurons (e.g. nodes). A plurality of neurons in a respective one of the plurality of hidden layers are connected with a plurality of neurons in an adjacent one of the plurality of hidden layers. Connects between neurons have different weights. The neural network has a structure that mimics a structure of a biological neural network. The neural network can solve problems using a non-deterministic manner.

SUMMARY

[0003] In one aspect, the present disclosure provides a computer-implemented method, comprising inputting a low-resolution image into a generator; and generating a high-resolution image using the generator based on the low-resolution image; wherein generating the high-resolution image comprises processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator, wherein a respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units.

[0004] Optionally, the respective one of the plurality of super-resolution generating units comprises a super-resolution generating block and a bilinear interpolation block; wherein generating the high-resolution image using the generator further comprises inputting a respective input to the super-resolution generating block and the bilinear interpolation block, respectively; processing the respective input using the super-resolution generating block to generate a first intermediate output from the super-resolution generating block; inputting the first intermediate output from the super-resolution generating block to the bilinear

interpolation block; and generating a second intermediate output using the bilinear interpolation block based on the respective input and the first intermediate output.

[0005] Optionally, the respective one of the plurality of super-resolution generating units further comprises a concatenation block; wherein generating the high-resolution image using the generator further comprises inputting the second intermediate output to the concatenation block; and generating the respective output from the concatenation block based on the second intermediate output.

[0006] Optionally, the super-resolution generating block comprises a plurality of encoding modules and a plurality of decoding modules arranged in a u-net architecture; a respective one of the plurality of encoding modules comprises a down-sampling convolutional layer, and a plurality of first residual blocks arranged in series; and a respective one of the plurality of decoding modules comprises a plurality of second residual blocks arranged in series, and an up-sampling convolutional layer.

[0007] Optionally, generating the high-resolution image using the generator further comprises inputting an output from a last one of the plurality of super-resolution generating units arranged in series into an additional super-resolution generating block; and processing the output from the last one of the plurality of super-resolution generating units using the additional super-resolution generating block to generate the high-resolution image.

[0008] Optionally, the computer-implemented method further comprises using a feedback loop to train the generator based on a generator loss function.

[0009] Optionally, the generator loss function is defined as $L_{total} = \sum_{i=0}^{n}(\alpha L_i^m + \beta L_i^{\varphi})$; wherein i stands for a respective number of a plurality of scales; L^m is a mean square error function; L^{φ} is a perceptual loss function; α stands for a weight coefficient for L^m; and β stands for a weight coefficients for L^{φ}.

[0010] Optionally, a neural network model is used to process a reference image and the high-resolution image; $L^m = min\sum_{j=0}^{N}\frac{1}{N}\left\|I_j - \hat{I}_j\right\|_2^2$; and $L_j^{\varphi}(\hat{I}, I) = \frac{1}{C_j H_j W_j}\left\|\varphi_j(\hat{I}) - \varphi_j(I)\right\|_2^2$; wherein I stands for the reference image; $\hat{I}$ stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of the

feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers.

[0011] Optionally, the neural network model is a VGG-19 model.

[0012] Optionally, the computer-implemented method further comprises pre-training a convolutional neural network by inputting a plurality of training low-resolution images into the convolutional neural network; inputting a plurality of training reference images into the convolutional neural network; generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images; computing loss for the plurality of training high-resolution images respectively based on the generator loss function; and tuning the convolutional neural network based on the loss.

[0013] Optionally, the computer-implemented method further comprises inputting a reference image and the high-resolution image into a discriminator; and verifying the high-resolution image using the discriminator based on the reference image; wherein the generator and the discriminator are components of a generative adversarial network (GAN).

[0014] Optionally, verifying the high-resolution image comprises processing the high-resolution image through a plurality of convolutional layers arranged in series, and a fully connection layer.

[0015] Optionally, the computer-implemented method further comprises using a feedback loop to mutually train the generator and the discriminator of the GAN, based on a generator loss function and a discriminator loss function.

[0016] Optionally, the generator loss function is defined as $\min V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_z(z)}\left[log\left(1 - D_i\left(G_i(z)\right)\right)\right] + \beta_i L_i^p$; and the discriminator loss function is defined as $\max V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_{data}(x)}\left[log\left(D_i(x)\right)\right] + E_{x\sim P_z(z)}\left[log\left(1 - D_i\left(G_i(z)\right)\right)\right]$; wherein D stands for the discriminator; G stands for the generator; i stands for a respective number of a plurality of scales; α_i and β_i stand for weight coefficients corresponding to the respective number of a plurality of scales; E stands for a probability density distribution; L_i^p is a perceptual loss function; $P_{data}(x)$ stands for a probability distribution of the reference image; x stands for a sample from $P_{data}(x)$; $P_z(z)$ stands for a probability distribution of the high-resolution image; and z stands for a sample from $P_z(z)$.

[0017] Optionally, a neural network model is used to process a reference image and the high-resolution image; the perceptual loss function is calculated as follows $L_j^{\varphi}(\hat{I}, I) = \frac{1}{C_j H_j W_j} \left\| \varphi_j(\hat{I}) - \varphi_j(I) \right\|_2^2$; wherein I stands for the reference image; Î stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of the feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers.

[0018] Optionally, the neural network model is a VGG-19 model.

[0019] Optionally, the neural network model is a sphereNet model.

[0020] Optionally, the computer-implemented method further comprises pre-training a convolutional neural network by inputting a plurality of training low-resolution images into the convolutional neural network; inputting a plurality of training reference images into the convolutional neural network; generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images; computing generator loss for the plurality of training high-resolution images respectively based on the generator loss function; computing discriminator loss for the plurality of training high-resolution images respectively based on the discriminator loss function; and tuning the convolutional neural network based on the generator loss and the discriminator loss.

[0021] In another aspect, the present disclosure provides an apparatus, comprising a memory; and one or more processors; wherein the memory and the one or more processors are connected with each other; and the memory stores computer-executable instructions for controlling the one or more processors to input a low-resolution image into a generator; and generate a high-resolution image using the generator based on the low-resolution image by processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator; wherein a respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units.

[0022] In another aspect, the present disclosure provides a computer-program product comprising a non-transitory tangible computer-readable medium having computer-readable instructions thereon, the computer-readable instructions being executable by a processor to cause the processor to perform inputting a low-resolution image into a generator; and generating a high-resolution image using the generator based on the low-resolution image; wherein generating the high-resolution image comprises processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator, wherein a respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units.

BRIEF DESCRIPTION OF THE FIGURES

[0023] The following drawings are merely examples for illustrative purposes according to various disclosed embodiments and are not intended to limit the scope of the present disclosure.

[0024] FIG. 1 is a flow chart illustrating a computer-implemented method in some embodiments according to the present disclosure.

[0025] FIG. 2 is a schematic diagram illustrating the structure of an apparatus for implementing a computer-implemented method in some embodiments according to the present disclosure.

[0026] FIG. 3 is a schematic diagram illustrating the structure of a generator in some embodiments according to the present disclosure.

[0027] FIG. 4 is a schematic diagram illustrating the structure of a respective one of the plurality of super-resolution generating units in some embodiments according to the present disclosure.

[0028] FIG. 5 is a schematic diagram illustrating the structure of a super-resolution generating block in some embodiments according to the present disclosure.

[0029] FIG. 6 is a schematic diagram illustrating the structure of a respective one of the plurality of encoding modules in some embodiments according to the present disclosure.

[0030] FIG. 7 is a schematic diagram illustrating the structure of a respective one of the plurality of decoding modules in some embodiments according to the present disclosure.

[0031] FIG. 8A shows a low-resolution image in some embodiments according to the present disclosure.

[0032] FIG. 8B shows a high-resolution image generated by a computer-implemented method in some embodiments according to the present disclosure.

[0033] FIG. 9 is a flow chart illustrating a computer-implemented method in some embodiments according to the present disclosure.

[0034] FIG. 10 is a schematic diagram illustrating the structure of an apparatus for implementing a computer-implemented method in some embodiments according to the present disclosure.

[0035] FIG. 11 is a schematic diagram illustrating the structure of a discriminator in some embodiments according to the present disclosure.

[0036] FIG. 12A shows a low-resolution image in some embodiments according to the present disclosure.

[0037] FIG. 12B shows a high-resolution image generated by a computer-implemented method in some embodiments according to the present disclosure.

[0038] FIG. 12C shows a reference image in some embodiments according to the present disclosure.

[0039] FIG. 13A shows a low-resolution image in some embodiments according to the present disclosure.

[0040] FIG. 13B shows a high-resolution image generated by a computer-implemented method in some embodiments according to the present disclosure.

[0041] FIG. 13C shows a reference image in some embodiments according to the present disclosure.

[0042] FIG. 14 is a schematic diagram of a structure of an apparatus in some embodiments according to the present disclosure.

DETAILED DESCRIPTION

[0043] The disclosure will now be described more specifically with reference to the following embodiments. It is to be noted that the following descriptions of some

embodiments are presented herein for purpose of illustration and description only. It is not intended to be exhaustive or to be limited to the precise form disclosed.

[0044] Various methods of enhancing image resolution have been developed in recent years. These methods include, for example, those utilizing a neural network. It is discovered in the present disclosure that these related methods typically require *a priori* knowledge in order to perform the downstream image enhancement process. In one example, a related method requires *a priori* semantic feature extraction to obtain an image parsing map and an image landmark heatmap. In another example, a related method requires *a priori* knowledge of a subject in the image, such as sex, age, and facial expression of the subject. These related methods involve time-consuming computation processes, and often times the results are not satisfactory.

[0045] Accordingly, the present disclosure provides, *inter alia*, a computer-implemented method, an apparatus, and a computer-program product that substantially obviate one or more of the problems due to limitations and disadvantages of the related art. In one aspect, the present disclosure provides a computer-implemented method. In some embodiments, the computer-implemented method includes inputting a low-resolution image into a generator; and generating a high-resolution image using the generator based on the low-resolution image. Optionally, the step of generating the high-resolution image includes processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator. Optionally, a respective output from the respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units. The present method does not require *a priori* knowledge such as *a priori* semantic feature extraction or *a priori* knowledge input. As a result, the present method can generate high resolution images without a time-consuming computation process. Enhanced images with super-high definition and excellent detail restoration can be obtained by the present method.

[0046] FIG. 1 is a flow chart illustrating a computer-implemented method in some embodiments according to the present disclosure. Referring to FIG. 1, the computer-implemented method in some embodiments includes inputting a low-resolution image into a generator; and generating a high-resolution image using the generator based on the low-resolution image. FIG. 2 is a schematic diagram illustrating the structure of an apparatus for implementing a computer-implemented method in some embodiments according to the

present disclosure. Referring to FIG. 2, the apparatus for implementing a computer-implemented method in some embodiments includes a generator 10, a loss function calculator 20, and a feedback loop 30. The generator 10 is configured to receive a low-resolution image as an input, and output a high-resolution image to the loss function calculator 20. The loss function calculator 20 in some embodiments computes a generator loss function based on the high-resolution image. The feedback loop 30 is configured to train the generator 10 based on a generator loss function.

[0047] FIG. 3 is a schematic diagram illustrating the structure of a generator in some embodiments according to the present disclosure. Referring to FIG. 3, the generator 10 in some embodiments includes a plurality of super-resolution generating units 100 arranged in series. Optionally, the generator 10 includes N number of super-resolution generating units, wherein $N \geq 2$. FIG. 3 shows an example of the generator 10 in which a total number of the plurality of super-resolution generating units 100 is three. Optionally, the generator 10 further includes an additional super-resolution generating block 200. Optionally, the additional super-resolution generating block 200 is configured to receive an input from a last one of the plurality of super-resolution generating units 100 arranged in series.

[0048] Accordingly, in some embodiments, the step of generating the high-resolution image includes processing the low-resolution image through a plurality of super-resolution generating units 100 arranged in series in the generator 10. Optionally, a respective output from a respective one of the plurality of super-resolution generating units 100 has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units 100. For example, each time the image is processes by one of the plurality of super-resolution generating units 100, the image resolution is increased by n times, wherein $n > 1$. Optionally, $n = 2$.

[0049] FIG. 4 is a schematic diagram illustrating the structure of a respective one of the plurality of super-resolution generating units in some embodiments according to the present disclosure. Referring to FIG. 4, the respective one of the plurality of super-resolution generating units 100 in some embodiments includes a super-resolution generating block 110, a bilinear interpolation block 120, and a concatenation block 130. Accordingly, in some embodiments, the step of generating the high-resolution image using the generator further includes inputting a respective input to the super-resolution generating block 110 and the bilinear interpolation block 120, respectively; processing the respective input using the super-

resolution generating block 110 to generate a first intermediate output from the super-resolution generating block 110; inputting the first intermediate output from the super-resolution generating block 110 to the bilinear interpolation block 120; generating a second intermediate output using the bilinear interpolation block 120 based on the respective input and the first intermediate output; inputting the second intermediate output to the concatenation block 130; and generating the respective output from the concatenation block 130 based on the second intermediate output.

[0050] In some embodiments, the bilinear interpolation block 120 performs a bilinear interpolation process. As used herein, bilinear interpolation refers to an interpolation method in which linear interpolation is two-dimensionally enlarged and an interpolation value weighted and averaged in accordance with a distance from each of the four pixels around one pixel to the interpolation object pixel is taken as a pixel value of the interpolation object pixel. In one example, the bilinear interpolation block 120 is a 2x bilinear interpolation block, in which the linear interpolation is two times enlarged.

[0051] FIG. 5 is a schematic diagram illustrating the structure of a super-resolution generating block in some embodiments according to the present disclosure. Referring to FIG. 5, in some embodiments, the super-resolution generating block 110 includes a plurality of encoding modules 1110 and a plurality of decoding modules 1120 arranged in a U-Net architecture. U-Net is a fully convolutional network modified and extended to work with fewer training images and to yield more precise segmentations. The U-Net architecture includes a contracting path and an expansive path, which together give the network a u-shaped architecture (see, e.g., Olaf Ronneberger et al., "U-Net: Convolutional Networks for Biomedical Image Segmentation," Medical Image Computing and Computer-Assisted Intervention (MICCAI) 9351, pages 234-241 (2015); the entire contents of which is hereby incorporated by reference).

[0052] FIG. 6 is a schematic diagram illustrating the structure of a respective one of the plurality of encoding modules in some embodiments according to the present disclosure. Referring to FIG. 6, in some embodiments, a respective one of the plurality of encoding modules 1110 includes a down-sampling convolutional layer 1111, and a plurality of first residual blocks 1112 arranged in series. In one example, the down-sampling convolutional layer 1111 is a 2x down-sampling convolutional layer. Optionally, the down-sampling convolutional layer 1111 includes a convolutional layer (e.g., a 5*5 convolutional layer with

a stride of 2) followed by a leaky rectified linear unit (LeakyReLU; see, e.g., Maas, et al., "Rectifier nonlinearities improve neural network acoustic models", In Proc. ICML, volume 30, 2013; and Xu, et al., "Empirical evaluation of rectified activations in convolutional network", arXiv preprint arXiv: 1505.00853, (2015); the entire contents of which is hereby incorporated by reference). Optionally, a respective one of the plurality of first residual blocks 1112 includes a convolutional layer (e.g., a 5*5 convolutional layer with a stride of 2), followed by a leaky rectified linear unit, then followed by another convolutional layer (e.g., a 5*5 convolutional layer with a stride of 2), and again followed by another leaky rectified linear unit.

[0053] As used herein, the term "convolution" refers to a process of processing an image. A convolutional kernel is used for a convolution. For example, each pixel of an input image has a value, a convolutional kernel starts at one pixel of the input image and moves over each pixel in an input image sequentially. At each position of the convolutional kernel, the convolutional kernel overlaps a few pixels on the image based on the size of the convolutional kernel. At a position of the convolutional kernel, a value of one of the few overlapped pixels is multiplied by a respective one value of the convolutional kernel to obtain a multiplied value of one of the few overlapped pixels. Subsequently, all multiplied values of the overlapped pixels are added to obtain a sum corresponding to the position of the convolutional kernel on the input image. By moving the convolutional kernel over each pixel of the input image, all the sums corresponding to all the position of the convolutional kernel are collected and output to form an output image. In one example, a convolution may extract different features of the input image using different convolutional kernels. In another example, a convolution process may add more features to the input image using different convolutional kernels.

[0054] As used herein, the term "convolutional layer" refers to a layer in a convolutional neural network. The convolutional layer is used to perform convolution on an input image to obtain an output image. Optionally, different convolutional kernels are used to performed different convolutions on the same input image. Optionally, different convolutional kernels are used to performed convolutions on different parts of the same input image. Optionally, different convolutional kernels are used to perform convolutions on different input images, for example, multiple images are inputted in a convolutional layer, a respective convolutional kernel is used to perform a convolution on an image of the multiple images. Optionally, different convolutional kernels are used according to different situations of the input image.

[0055] FIG. 7 is a schematic diagram illustrating the structure of a respective one of the plurality of decoding modules in some embodiments according to the present disclosure. Referring to FIG. 7, in some embodiments, a respective one of the plurality of decoding modules 1120 includes a plurality of second residual blocks 1121 arranged in series, and an up-sampling convolutional layer 1122. In one example, the up-sampling convolutional layer 1122 is a 2x down-sampling convolutional layer. Optionally, the up-sampling convolutional layer 1122 includes a de-convolutional layer (e.g., a 5*5 de-convolutional layer with a stride of 2) followed by a leaky rectified linear unit. Optionally, a respective one of the plurality of second residual blocks 1121 includes a convolutional layer (e.g., a 5*5 convolutional layer with a stride of 2), followed by a leaky rectified linear unit, then followed by another convolutional layer (e.g., a 5*5 convolutional layer with a stride of 2), and again followed by another leaky rectified linear unit.

[0056] As used herein, the term "down-sampling" refers to a process of extracting features of an input image, and outputting an output image with a smaller scale. As used herein, the term "up-sampling" refers to a process of adding more information to an input image, and outputting an outputting image with a larger scale. As used herein, the term "scale" refers to one or any combinations of three dimensions of an image, including one or any combinations of a width of the image, a height of the image, and a depth of the image. In one example, the scale of an image (e.g., a feature map, a data, a signal) refers to a "volume" of an image, which includes the width of the image, the height of the image, and the depth of the image. In another example, spatial scale of an image (e.g., a feature map, a data, a signal) refers to a width and length of the image, e.g., width × length. As used herein, the term "feature" generally refers to information about an image or a portion of an image, including any localized optical characteristic in an image, e.g. a spot, a line, or a pattern. In one exemplary instance, a feature takes the form of spatial information defined by a vector and/or integer.

[0057] Referring to FIG. 3, in some embodiments, the generator 10 further includes an additional super-resolution generating block 200. Optionally, the additional super-resolution generating block 200 is configured to receive an input from a last one of the plurality of super-resolution generating units 100 arranged in series. Accordingly, in some embodiments, the step of generating the high-resolution image further includes inputting an output from a last one of the plurality of super-resolution generating units 100 arranged in series into an additional super-resolution generating block 200; and processing the output from the last one

of the plurality of super-resolution generating units 100 using the additional super-resolution generating block 200 to generate the high-resolution image.

[0058] Optionally, the additional super-resolution generating block 200 has an architecture similar or identical to a respective one of the plurality of super-resolution generating units 100 (e.g., those illustrated in FIGs. 5 to 7).

[0059] In some embodiments, the computer-implemented method further includes using a feedback loop to train the generator based on a generator loss function. As used herein, the term "loss function" refer to a functions that indicates loss errors. In one example, the generator loss function is defined as $L_{total} = \sum_{i=0}^{n}(\alpha L_i^m + \beta L_i^{\varphi})$; wherein i stands for a respective number of a plurality of scales; L^m is a mean square error function; L^{φ} is a perceptual loss function; α stands for a weight coefficient for L^m; and β stands for a weight coefficients for L^{φ}. Optionally, a neural network model (such as a convolutional neural network) is used to process a reference image and the high-resolution image. Optionally, the $L^m = min \sum_{j=0}^{N} \frac{1}{N} \left\| I_j - \hat{I}_j \right\|_2^2$; and $L_j^{\varphi}(\hat{I}, I) = \frac{1}{C_j H_j W_j} \left\| \varphi_j(\hat{I}) - \varphi_j(I) \right\|_2^2$; wherein I stands for the reference image; $\hat{I}$ stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of the feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers. Optionally, the neural network model is a VGG-19 model (e.g., a VGG-19 model originally developed by Visual Geometry Group of University of Oxford).

[0060] As used herein, the term "feature map" refers to a map or data representing a particular feature or parameter or characteristic of an image. The feature map may be graphically or mathematically represented. The feature map may be a form of simplified or alternative representation of an image. For example, the feature map is an outcome of applying a function to a topologically arranged vector of numbers to obtain a vector of corresponding output numbers preserving a topology. For example, a "feature map" is the result of using a layer of convolutional neural network to process an image or another feature map, for example, an image of scale (28, 28, 1) is inputted into the convolution layer, and the convolutional layer having 23 number of kernels with a kernel size of 1*1 generates a feature map of scale (26, 26, 32) by computing 32 kernels over the input image. A feature map has a

width W, a height H, and a depth D, for example, the feature map of scale (26, 26, 32) has a width of 26, a height of 26, and a depth of 32. The feature map has 32 feature sub-maps, a total number of the feature sub-maps is equivalent to the depth of the feature map. The depth D is also represented by channels of the feature map, so the feature map of scale (26, 26, 32) includes 32 channels and each channels has a 26×26 grid of values.

[0061] As used herein, the term "neural network" refers to a network used for solving artificial intelligence (AI) problems. A neural network includes a plurality of hidden layers. A respective one of the plurality of hidden layers includes a plurality of neurons (e.g. nodes). A plurality of neurons in a respective one of the plurality of hidden layers are connected with a plurality of neurons in an adjacent one of the plurality of hidden layers. Connects between neurons have different weights. The neural network has a structure that mimics a structure of a biological neural network. The neural network can solve problems using a non-deterministic manner. Parameters of the neural network can be tuned by pre-training, for example, a large amount of problems are input in the neural network, and results are obtained from the neural network. Feedbacks on these results is fed back into the neural network to allow the neural network to tune the parameters of the neural network. The pre-training allows the neural network to have a stronger problem-solving ability.

[0062] As used herein, the term "convolutional neural network" refers to a deep feed-forward artificial neural network. Optionally, a convolutional neural network includes a plurality of convolutional layers, a plurality of up-sampling layers, and a plurality of down-sampling layers. For example, a respective one of the plurality of convolutional layers can process an image. An up-sampling layer and a down-sampling layer can change a scale of an input image to one corresponding to a certain convolutional layer. The output from the up-sampling layer or the down-sampling layer can then be processed by a convolutional layer of a corresponding scale. This enables the convolutional layer to add or extract a feature having a scale different from that of the input image. By pre-training, parameters include, but are not limited to, a convolutional kernel, a bias, and a weight of a convolutional layer of a convolutional neural network can be tuned. Accordingly, the convolutional neural network can be used in various applications such as image recognition, image feature extraction, and image feature addition.

[0063] In one example, the computer-implemented method is used for de-blurring a low-resolution image. FIG. 8A shows a low-resolution image in some embodiments according to

the present disclosure. FIG. 8B shows a high-resolution image generated by a computer-implemented method in some embodiments according to the present disclosure. As shown in FIG. 8B, the present method can de-blur a low resolution image to a high-resolution image with a relatively high degree of detail restoration.

[0064] In some embodiments, the computer-implemented method further includes pre-training a convolutional neural network by inputting a plurality of training low-resolution images into the convolutional neural network; inputting a plurality of training reference images into the convolutional neural network; generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images; computing loss for the plurality of training high-resolution images respectively based on the generator loss function; and tuning the convolutional neural network based on the loss. In one example, the present training method uses a database having a large number of high-resolution images (e.g., 4000 frames of natural scene images). The training learning rate may be set to 1 e^{-4}, with a number of iterations set to 4000. In another example, the convolutional neural network has converged after about 80 iterations.

[0065] FIG. 9 is a flow chart illustrating a computer-implemented method in some embodiments according to the present disclosure. Referring to FIG. 9, in some embodiments, the computer-implemented method includes inputting a low-resolution image into a generator; generating a high-resolution image using the generator based on the low-resolution image; inputting a reference image and the high-resolution image into a discriminator; and verifying the high-resolution image using the discriminator based on the reference image. Optionally, the generator and the discriminator are components of a generative adversarial network (GAN). As used herein, the term "generative adversarial network" refers to a neural network that includes a generator and a discriminator. In one example, the discriminator may be initially trained to distinguish between actual data and fake data that could be drawn from another random distribution. Optionally, the initial training of the discriminator is not mandatory, and the discriminator could be trained in step with the generator. Often for reasons of efficiency and stability of training, a trained discriminator is used initially and training is continued with the generator. The generator may be initialized with a random distribution of data samples (for example, a normal distribution). During training of the generator, a fake sample (or a batch of fake samples) is randomly drawn from the generator distribution and is passed to the discriminator. An actual data sample (or a batch of actual data samples) are also passed to the discriminator. The discriminator in turns tries to

discriminate between these samples as true or fake. A 'true sample' is one drawn from the actual data and a 'fake sample' is one generated by the generator. Depending on the output of discriminator, a loss function is minimized at the generator. As such, the parameters (e.g. model weights) of the generator are adjusted, which in turn allows the generator to better discriminate between true and fake samples.

[0066] In some embodiments, the step of generating the high-resolution image includes processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator, wherein a respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units; and the step of verifying the high-resolution image includes processing the high-resolution image through a plurality of convolutional layers arranged in series, and a fully connection layer.

[0067] FIG. 10 is a schematic diagram illustrating the structure of an apparatus for implementing a computer-implemented method in some embodiments according to the present disclosure. Referring to FIG. 10, the apparatus for implementing a computer-implemented method in some embodiments includes a generator 10, a loss function calculator 20, a discriminator 40, and a feedback loop 30. The generator 10 is configured to receive a low-resolution image as an input, and output a high-resolution image to the loss function calculator 20. In some embodiments, the generator 10 further outputs the high-resolution image to the discriminator 40. The discriminator 40 is configured to receive a reference image (e.g., a high-resolution actual image, e.g., "true sample"), and determine whether the high-resolution image is real or fake. The loss function calculator 20 in some embodiments computes a generator loss function and a discriminator loss function based on the high-resolution image and the reference image. The feedback loop 30 is configured to mutually train the generator 10 and the discriminator 40 of the GAN, based on a generator loss function and a discriminator loss function. In one example, the structure of the generator 10 is that described above, e.g., in connection with FIGs. 3 to 7 and associated descriptions.

[0068] FIG. 11 is a schematic diagram illustrating the structure of a discriminator in some embodiments according to the present disclosure. Referring to FIG. 11, in some embodiments, the discriminator 40 includes a plurality of convolutional layers 300 arranged in series, and a fully connection layer 400. Optionally, the fully connection layer 400 is

configured to receive an input from a last one of the plurality of convolutional layers 300 arranged in series. In one example, a total number of the plurality of convolutional layers 300 is five. In another example, each of the plurality of convolutional layers 300 is a 5*5 convolutional layer with a stride of 2, followed by a rectified linear unit (ReLU; see, e.g., Nair, Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines", Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, (2010); the entire contents of which is hereby incorporated by reference). In another example, kernel sizes of the plurality of convolutional layers 300 are 32, 64, 128, 96, and 96, respectively. In another example, the fully connection layer 400 a 96*16*16 convolutional layer.

[0069] In some embodiments, the generator loss function is defined as $\min V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_z(z)} \left[log\left(1 - D_i\left(G_i(z)\right)\right)\right] + \beta_i L_i^p$; and the discriminator loss function is defined as $\max V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_{data}(x)}\left[log\left(D_i(x)\right)\right] + E_{x\sim P_z(z)}\left[log\left(1 - D_i\left(G_i(z)\right)\right)\right]$; wherein D stands for the discriminator; G stands for the generator; i stands for a respective number of a plurality of scales; α_i and β_i stand for weight coefficients corresponding to the respective number of a plurality of scales; E stands for a probability density distribution; L_i^p is a perceptual loss function; $P_{data}(x)$ stands for a probability distribution of the reference image; x stands for a sample from $P_{data}(x)$; $P_z(z)$ stands for a probability distribution of the high-resolution image; and z stands for a sample from $P_z(z)$.

[0070] In some embodiments, a neural network model is used to process a reference image and the high-resolution image. Optionally, the perceptual loss function is calculated as follows: $L_j^{\varphi}\left(\hat{I}, I\right) = \frac{1}{C_j H_j W_j}\left\|\varphi_j\left(\hat{I}\right) - \varphi_j(I)\right\|_2^2$; wherein I stands for the reference image; Î stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of the feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers. Optionally, the neural network model is a VGG-19 model. Optionally, the neural network model is a sphereNet model (see, e.g., Coors et al., "SphereNet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images," European Conference on Computer Vision 2018: Computer Vision – ECCV 2018 pp 525-541; the entire contents of which is hereby incorporated by reference).

[0071] In one example, the computer-implemented method is used for face hallucination of a facial image of a subject. FIG. 12A shows a low-resolution image in some embodiments according to the present disclosure. FIG. 12B shows a high-resolution image generated by a computer-implemented method in some embodiments according to the present disclosure. FIG. 12C shows a reference image in some embodiments according to the present disclosure. As shown in FIG. 12B, the present method can enhance the facial image to a high resolution with details of the face (e.g., hair, eyebrows, glasses, wrinkles, and so on) restored.

[0072] In one example, the computer-implemented method is used for de-blurring a low-resolution image. FIG. 13A shows a low-resolution image in some embodiments according to the present disclosure. FIG. 13B shows a high-resolution image generated by a computer-implemented method in some embodiments according to the present disclosure. FIG. 13C shows a reference image in some embodiments according to the present disclosure. As shown in FIG. 13B, the present method can de-blur a low resolution image to a high-resolution image with a relatively high degree of detail restoration.

[0073] In some embodiments, the computer-implemented method further includes pre-training a convolutional neural network by inputting a plurality of training low-resolution images into the convolutional neural network; inputting a plurality of training reference images into the convolutional neural network; generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images; computing generator loss for the plurality of training high-resolution images respectively based on the generator loss function; computing discriminator loss for the plurality of training high-resolution images respectively based on the discriminator loss function; and tuning the convolutional neural network based on the generator loss and the discriminator loss. In one example, the present training method uses a database having a large number of high-resolution images (e.g., 4000 frames of natural scene images, or a celebA-HQ facial image database with 200K frames of facial images). The training learning rate may be set to 1 e^{-4}, with a number of iterations set to 100.

[0074] In another aspect, the present disclosure provides an apparatus. FIG. 14 is a schematic diagram of a structure of an apparatus in some embodiments according to the present disclosure. Referring to FIG.14, in some embodiments, the apparatus includes the central processing unit (CPU) configured to perform actions according to the computer-executable instructions stored in a ROM or in a RAM. Optionally, data and programs

required for a computer system are stored in RAM. Optionally, the CPU, the ROM, and the RAM are electrically connected to each other via bus. Optionally, an input/output interface is electrically connected to the bus.

[0075] In some embodiments, the apparatus includes a memory, and one or more processors, wherein the memory and the one or more processors are connected with each other. In some embodiments, the memory stores computer-executable instructions for controlling the one or more processors to input a low-resolution image into a generator; and generate a high-resolution image using the generator based on the low-resolution image by processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator. Optionally, a respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units.

[0076] In some embodiments, the respective one of the plurality of super-resolution generating units includes a super-resolution generating block, a bilinear interpolation block, and a concatenation block. Optionally, the memory further stores computer-executable instructions for controlling the one or more processors to input a respective input to the super-resolution generating block and the bilinear interpolation block, respectively; process the respective input using the super-resolution generating block to generate a first intermediate output from the super-resolution generating block; input the first intermediate output from the super-resolution generating block to the bilinear interpolation block; generate a second intermediate output using the bilinear interpolation block based on the respective input and the first intermediate output; input the second intermediate output to the concatenation block; and generate the respective output from the concatenation block based on the second intermediate output.

[0077] In some embodiments, the super-resolution generating block includes a plurality of encoding modules and a plurality of decoding modules arranged in a u-net architecture. Optionally, a respective one of the plurality of encoding modules includes a down-sampling convolutional layer, and a plurality of first residual blocks arranged in series. Optionally, a respective one of the plurality of decoding modules includes a plurality of second residual blocks arranged in series, and an up-sampling convolutional layer.

[0078] In some embodiments, the memory further stores computer-executable instructions for controlling the one or more processors to input an output from a last one of the plurality of super-resolution generating units arranged in series into an additional super-resolution generating block; and process the output from the last one of the plurality of super-resolution generating units using the additional super-resolution generating block to generate the high-resolution image.

[0079] In some embodiments, the memory further stores computer-executable instructions for controlling the one or more processors to use a feedback loop to train the generator based on a generator loss function. Optionally, the generator loss function is defined as $L_{total} = \sum_{i=0}^{n}(\alpha L_i^m + \beta L_i^{\varphi})$; wherein i stands for a respective number of a plurality of scales; L^m is a mean square error function; L^{φ} is a perceptual loss function; α stands for a weight coefficient for L^m; and β stands for a weight coefficients for L^{φ}.

[0080] In some embodiments, a neural network model is used to process a reference image and the high-resolution image. Optionally, $L^m = min \sum_{j=0}^{N} \frac{1}{N} \|I_j - \hat{I}_j\|_2^2$; and $L_j^{\varphi}(\hat{I}, I) = \frac{1}{C_j H_j W_j} \|\varphi_j(\hat{I}) - \varphi_j(I)\|_2^2$; wherein I stands for the reference image; $\hat{I}$ stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of the feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers. Optionally, the neural network model is a VGG-19 model.

[0081] In some embodiments, the memory further stores computer-executable instructions for controlling the one or more processors to pre-train a convolutional neural network by inputting a plurality of training low-resolution images into the convolutional neural network; inputting a plurality of training reference images into the convolutional neural network; generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images; computing loss for the plurality of training high-resolution images respectively based on the generator loss function; and tuning the convolutional neural network based on the loss.

[0082] In some embodiments, the memory further stores computer-executable instructions for controlling the one or more processors to input a reference image and the high-resolution

image into a discriminator; and verify the high-resolution image using the discriminator based on the reference image. Optionally, the generator and the discriminator are components of a generative adversarial network (GAN).

[0083] In some embodiments, the memory further stores computer-executable instructions for controlling the one or more processors to process the high-resolution image through a plurality of convolutional layers arranged in series, and a fully connection layer.

[0084] In some embodiments, the memory further stores computer-executable instructions for controlling the one or more processors to use a feedback loop to mutually train the generator and the discriminator of the GAN, based on a generator loss function and a discriminator loss function. Optionally, the generator loss function is defined as $\min V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_z(z)}\left[log\left(1 - D_i\left(G_i(z)\right)\right)\right] + \beta_i L_i^p$; and the discriminator loss function is defined as $\max V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_{data}(x)}\left[log\left(D_i(x)\right)\right] + E_{x\sim P_z(z)}\left[log\left(1 - D_i\left(G_i(z)\right)\right)\right]$; wherein D stands for the discriminator; G stands for the generator; i stands for a respective number of a plurality of scales; α_i and β_i stand for weight coefficients corresponding to the respective number of a plurality of scales; E stands for a probability density distribution; L_i^p is a perceptual loss function; $P_{data}(x)$ stands for a probability distribution of the reference image; x stands for a sample from $P_{data}(x)$; $P_z(z)$ stands for a probability distribution of the high-resolution image; and z stands for a sample from $P_z(z)$.

[0085] In some embodiments, a neural network model is used to process a reference image and the high-resolution image. Optionally, the perceptual loss function is calculated as follows $L_j^{\varphi}\left(\hat{I}, I\right) = \frac{1}{C_j H_j W_j}\left\|\varphi_j\left(\hat{I}\right) - \varphi_j(I)\right\|_2^2$; wherein I stands for the reference image; $\hat{I}$ stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of the feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers. Optionally, the neural network model is a VGG-19 model. Optionally, the neural network model is a sphereNet model.

[0086] In some embodiments, the memory further stores computer-executable instructions for controlling the one or more processors to pre-train a convolutional neural network by

inputting a plurality of training low-resolution images into the convolutional neural network; inputting a plurality of training reference images into the convolutional neural network; generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images; computing generator loss for the plurality of training high-resolution images respectively based on the generator loss function; computing discriminator loss for the plurality of training high-resolution images respectively based on the discriminator loss function; and tuning the convolutional neural network based on the generator loss and the discriminator loss.

[0087] In another aspect, the present disclosure provides a computer-program product including a non-transitory tangible computer-readable medium having computer-readable instructions thereon. In some embodiments, the computer-readable instructions being executable by a processor to cause the processor to perform inputting a low-resolution image into a generator; and generating a high-resolution image using the generator based on the low-resolution image. Optionally, generating the high-resolution image includes processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator, wherein a respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units.

[0088] In some embodiments, the respective one of the plurality of super-resolution generating units includes a super-resolution generating block, a bilinear interpolation block, and a concatenation block. Optionally, generating the high-resolution image using the generator further includes inputting a respective input to the super-resolution generating block and the bilinear interpolation block, respectively; processing the respective input using the super-resolution generating block to generate a first intermediate output from the super-resolution generating block; inputting the first intermediate output from the super-resolution generating block to the bilinear interpolation block; generating a second intermediate output using the bilinear interpolation block based on the respective input and the first intermediate output; inputting the second intermediate output to the concatenation block; and generating the respective output from the concatenation block based on the second intermediate output.

[0089] In some embodiments, the super-resolution generating block includes a plurality of encoding modules and a plurality of decoding modules arranged in a U-Net architecture. Optionally, a respective one of the plurality of encoding modules includes a down-sampling

convolutional layer, and a plurality of first residual blocks arranged in series; and a respective one of the plurality of decoding modules includes a plurality of second residual blocks arranged in series, and an up-sampling convolutional layer.

[0090] In some embodiments, generating the high-resolution image using the generator further includes inputting an output from a last one of the plurality of super-resolution generating units arranged in series into an additional super-resolution generating block; and processing the output from the last one of the plurality of super-resolution generating units using the additional super-resolution generating block to generate the high-resolution image.

[0091] In some embodiments, the computer-readable instructions are executable by a processor to cause the processor to further perform using a feedback loop to train the generator based on a generator loss function. Optionally, the generator loss function is defined as $L_{total} = \sum_{i=0}^{n}(\alpha L_i^m + \beta L_i^{\varphi})$; wherein i stands for a respective number of a plurality of scales; L^m is a mean square error function; L^{φ} is a perceptual loss function; α stands for a weight coefficient for L^m; and β stands for a weight coefficients for L^{φ}.

[0092] In some embodiments, a neural network model is used to process a reference image and the high-resolution image. Optionally, $L^m = min \sum_{j=0}^{N} \frac{1}{N} \|I_j - \hat{I}_j\|_2^2$; and $L_j^{\varphi}(\hat{I}, I) = \frac{1}{C_j H_j W_j} \|\varphi_j(\hat{I}) - \varphi_j(I)\|_2^2$; wherein I stands for the reference image; $\hat{I}$ stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of the feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers. Optionally, the neural network model is a VGG-19 model.

[0093] In some embodiments, the computer-readable instructions are executable by a processor to cause the processor to further perform pre-training a convolutional neural network by inputting a plurality of training low-resolution images into the convolutional neural network; inputting a plurality of training reference images into the convolutional neural network; generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images; computing loss for the plurality of training high-resolution images respectively based on the generator loss function; and tuning the convolutional neural network based on the loss.

[0094] In some embodiments, the computer-readable instructions are executable by a processor to cause the processor to further perform inputting a reference image and the high-resolution image into a discriminator; and verifying the high-resolution image using the discriminator based on the reference image. Optionally, the generator and the discriminator are components of a generative adversarial network (GAN).

[0095] In some embodiments, verifying the high-resolution image includes processing the high-resolution image through a plurality of convolutional layers arranged in series, and a fully connection layer.

[0096] In some embodiments, the computer-readable instructions are executable by a processor to cause the processor to further perform using a feedback loop to mutually train the generator and the discriminator of the GAN, based on a generator loss function and a discriminator loss function. Optionally, the generator loss function is defined as $\min V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_z(z)} \left[log\left(1 - D_i\left(G_i(z)\right)\right)\right] + \beta_i L_i^p$; and the discriminator loss function is defined as $\max V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_{data}(x)}\left[log\left(D_i(x)\right)\right] + E_{x\sim P_z(z)}\left[log\left(1 - D_i\left(G_i(z)\right)\right)\right]$; wherein D stands for the discriminator; G stands for the generator; i stands for a respective number of a plurality of scales; α_i and β_i stand for weight coefficients corresponding to the respective number of a plurality of scales; E stands for a probability density distribution; L_i^p is a perceptual loss function; $P_{data}(x)$ stands for a probability distribution of the reference image; x stands for a sample from $P_{data}(x)$; $P_z(z)$ stands for a probability distribution of the high-resolution image; and z stands for a sample from $P_z(z)$.

[0097] In some embodiments, a neural network model is used to process a reference image and the high-resolution image. Optionally, the perceptual loss function is calculated as follows $L_j^{\varphi}\left(\hat{I}, I\right) = \frac{1}{C_j H_j W_j}\left\|\varphi_j\left(\hat{I}\right) - \varphi_j(I)\right\|_2^2$; wherein I stands for the reference image; Î stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of the feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers. Optionally, the neural network model is a VGG-19 model. Optionally, the neural network model is a sphereNet model.

[0098] In some embodiments, the computer-readable instructions are executable by a processor to cause the processor to further perform pre-training a convolutional neural network by inputting a plurality of training low-resolution images into the convolutional neural network; inputting a plurality of training reference images into the convolutional neural network; generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images; computing generator loss for the plurality of training high-resolution images respectively based on the generator loss function; computing discriminator loss for the plurality of training high-resolution images respectively based on the discriminator loss function; and tuning the convolutional neural network based on the generator loss and the discriminator loss.

[0099] Various illustrative neural networks, units, modules, and other operations described in connection with the configurations disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. Such neural networks, units, modules, and other operations may be implemented or performed with a general purpose processor, a digital signal processor (DSP), an ASIC or ASSP, an FPGA or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to produce the configuration as disclosed herein. For example, such a configuration may be implemented at least in part as a hard-wired circuit, as a circuit configuration fabricated into an application-specific integrated circuit, or as a firmware program loaded into non-volatile storage or a software program loaded from or into a data storage medium as machine-readable code, such code being instructions executable by an array of logic elements such as a general purpose processor or other digital signal processing unit. A general purpose processor may be a microprocessor, but in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration. A software module may reside in a non-transitory storage medium such as RAM (random-access memory), ROM (read-only memory), nonvolatile RAM (NVRAM) such as flash RAM, erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), registers, hard disk, a removable disk, or a CD-ROM; or in any other form of storage medium known in the art. An illustrative storage medium is coupled to the processor such the processor can read information from, and write information to, the storage medium.

In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a user terminal.

[0100] The foregoing description of the embodiments of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form or to exemplary embodiments disclosed. Accordingly, the foregoing description should be regarded as illustrative rather than restrictive. Obviously, many modifications and variations will be apparent to practitioners skilled in this art. The embodiments are chosen and described in order to explain the principles of the invention and its best mode practical application, thereby to enable persons skilled in the art to understand the invention for various embodiments and with various modifications as are suited to the particular use or implementation contemplated. It is intended that the scope of the invention be defined by the claims appended hereto and their equivalents in which all terms are meant in their broadest reasonable sense unless otherwise indicated. Therefore, the term "the invention", "the present invention" or the like does not necessarily limit the claim scope to a specific embodiment, and the reference to exemplary embodiments of the invention does not imply a limitation on the invention, and no such limitation is to be inferred. The invention is limited only by the spirit and scope of the appended claims. Moreover, these claims may refer to use "first", "second", etc. following with noun or element. Such terms should be understood as a nomenclature and should not be construed as giving the limitation on the number of the elements modified by such nomenclature unless specific number has been given. Any advantages and benefits described may not apply to all embodiments of the invention. It should be appreciated that variations may be made in the embodiments described by persons skilled in the art without departing from the scope of the present invention as defined by the following claims. Moreover, no element and component in the present disclosure is intended to be dedicated to the public regardless of whether the element or component is explicitly recited in the following claims.

WHAT IS CLAIMED IS:

1. A computer-implemented method, comprising:

inputting a low-resolution image into a generator; and

generating a high-resolution image using the generator based on the low-resolution image;

wherein generating the high-resolution image comprises processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator, wherein a respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units.

2. The computer-implemented method of claim 1, wherein the respective one of the plurality of super-resolution generating units comprises a super-resolution generating block and a bilinear interpolation block;

wherein generating the high-resolution image using the generator further comprises:

inputting a respective input to the super-resolution generating block and the bilinear interpolation block, respectively;

processing the respective input using the super-resolution generating block to generate a first intermediate output from the super-resolution generating block;

inputting the first intermediate output from the super-resolution generating block to the bilinear interpolation block; and

generating a second intermediate output using the bilinear interpolation block based on the respective input and the first intermediate output.

3. The computer-implemented method of claim 2, wherein the respective one of the plurality of super-resolution generating units further comprises a concatenation block;

wherein generating the high-resolution image using the generator further comprises:

inputting the second intermediate output to the concatenation block; and

generating the respective output from the concatenation block based on the second intermediate output.

4. The computer-implemented method of claim 2 or claim 3, wherein the super-resolution generating block comprises a plurality of encoding modules and a plurality of decoding modules arranged in a u-net architecture;

a respective one of the plurality of encoding modules comprises a down-sampling convolutional layer, and a plurality of first residual blocks arranged in series; and

a respective one of the plurality of decoding modules comprises a plurality of second residual blocks arranged in series, and an up-sampling convolutional layer.

5. The computer-implemented method of any one of claims 1 to 4, wherein generating the high-resolution image using the generator further comprises:

inputting an output from a last one of the plurality of super-resolution generating units arranged in series into an additional super-resolution generating block; and

processing the output from the last one of the plurality of super-resolution generating units using the additional super-resolution generating block to generate the high-resolution image.

6. The computer-implemented method of any one of claims 1 to 5, the computer-implemented method further comprising using a feedback loop to train the generator based on a generator loss function.

7. The computer-implemented method of claim 6, wherein the generator loss function is defined as:

$$L_{total} = \sum_{i=0}^{n}(\alpha L_i^m + \beta L_i^{\varphi});$$

wherein i stands for a respective number of a plurality of scales; L^m is a mean square error function; L^{φ} is a perceptual loss function; α stands for a weight coefficient for L^m; and β stands for a weight coefficients for L^{φ}.

8. The computer-implemented method of claim 7, wherein a neural network model is used to process a reference image and the high-resolution image;

$$L^m = min \sum_{j=0}^{N} \frac{1}{N} \left\| I_j - \hat{I}_j \right\|_2^2; \text{ and}$$

$$L_j^{\varphi}(\hat{I}, I) = \frac{1}{C_j H_j W_j} \left\| \varphi_j(\hat{I}) - \varphi_j(I) \right\|_2^2;$$

wherein I stands for the reference image; Î stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of

the feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers.

9. The computer-implemented method of claim 8, wherein the neural network model is a VGG-19 model.

10. The computer-implemented method of any one of claims 6 to 9, further comprising pre-training a convolutional neural network by:

inputting a plurality of training low-resolution images into the convolutional neural network;

inputting a plurality of training reference images into the convolutional neural network;

generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images;

computing loss for the plurality of training high-resolution images respectively based on the generator loss function; and

tuning the convolutional neural network based on the loss.

11. The computer-implemented method of any one of claims 1 to 10, further comprising:

inputting a reference image and the high-resolution image into a discriminator; and

verifying the high-resolution image using the discriminator based on the reference image;

wherein the generator and the discriminator are components of a generative adversarial network.

12. The computer-implemented method of claim 11, wherein verifying the high-resolution image comprises processing the high-resolution image through a plurality of convolutional layers arranged in series, and a fully connection layer.

13. The computer-implemented method of claim 11 or claim 12, further comprising using a feedback loop to mutually train the generator and the discriminator of the generative adversarial network, based on a generator loss function and a discriminator loss function.

14. The computer-implemented method of claim 13, wherein the generator loss function is defined as:

$$\min V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_z(z)}\left[log\left(1 - D_i\left(G_i(z)\right)\right)\right] + \beta_i L_i^p;$$ and

the discriminator loss function is defined as:

$$\max V(D,G) = \sum_{i=0}^{3} \alpha_i E_{x\sim P_{data}(x)}\left[log\left(D_i(x)\right)\right] + E_{x\sim P_z(z)}\left[log\left(1 - D_i\left(G_i(z)\right)\right)\right];$$

wherein D stands for the discriminator; G stands for the generator; i stands for a respective number of a plurality of scales; α_i and β_i stand for weight coefficients corresponding to the respective number of a plurality of scales; E stands for a probability density distribution; L_i^p is a perceptual loss function; $P_{data}(x)$ stands for a probability distribution of the reference image; x stands for a sample from $P_{data}(x)$; $P_z(z)$ stands for a probability distribution of the high-resolution image; and z stands for a sample from $P_z(z)$.

15. The computer-implemented method of claim 14, wherein a neural network model is used to process a reference image and the high-resolution image;

the perceptual loss function is calculated as follows:

$$L_j^{\varphi}\left(\hat{I}, I\right) = \frac{1}{C_j H_j W_j}\left\|\varphi_j\left(\hat{I}\right) - \varphi_j(I)\right\|_2^2;$$

wherein I stands for the reference image; $\hat{I}$ stands for the high-resolution image; φ stands for the neural network model; j stands for a respective one of a plurality of output layers of the neural network model; C stands for a number of channels of a feature map outputted from the respective one of a plurality of output layers; H stands for a height of the feature map outputted from the respective one of a plurality of output layers; and W stands for a width of the feature map outputted from the respective one of a plurality of output layers.

16. The computer-implemented method of claim 15, wherein the neural network model is a VGG-19 model.

17. The computer-implemented method of claim 15, wherein the neural network model is a sphereNet model.

18. The computer-implemented method of any one of claims 13 to 17, further comprising pre-training a convolutional neural network by:

inputting a plurality of training low-resolution images into the convolutional neural network;

inputting a plurality of training reference images into the convolutional neural network;

generating a plurality of training high-resolution images using the generator respectively based on the plurality of training low-resolution images;

computing generator loss for the plurality of training high-resolution images respectively based on the generator loss function;

computing discriminator loss for the plurality of training high-resolution images respectively based on the discriminator loss function; and

tuning the convolutional neural network based on the generator loss and the discriminator loss.

19. An apparatus, comprising:

a memory; and

one or more processors;

wherein the memory and the one or more processors are connected with each other; and

the memory stores computer-executable instructions for controlling the one or more processors to:

input a low-resolution image into a generator; and

generate a high-resolution image using the generator based on the low-resolution image by processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator;

wherein a respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units.

20. A computer-program product comprising a non-transitory tangible computer-readable medium having computer-readable instructions thereon, the computer-readable instructions being executable by a processor to cause the processor to perform:

inputting a low-resolution image into a generator; and

generating a high-resolution image using the generator based on the low-resolution image;

wherein generating the high-resolution image comprises processing the low-resolution image through a plurality of super-resolution generating units arranged in series in the generator, wherein a respective output from a respective one of the plurality of super-resolution generating units has a respective increased image resolution as compared to a respective input to the respective one of the plurality of super-resolution generating units.

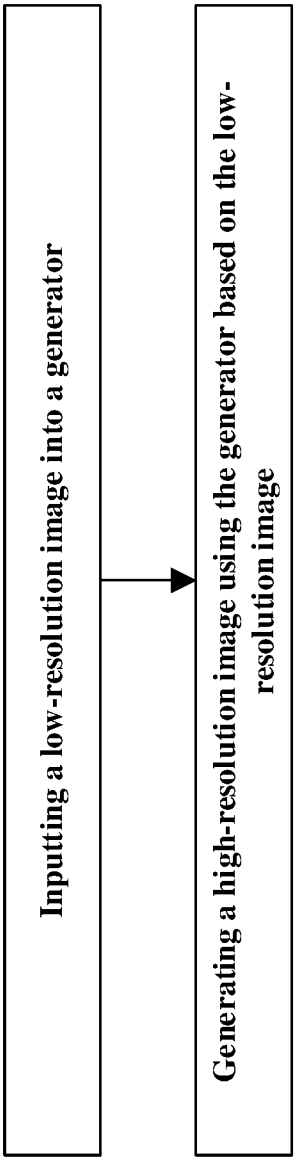

Inputting a low-resolution image into a generator
Generating a high-resolution image using the generator based on the low-resolution image

FIG. 1

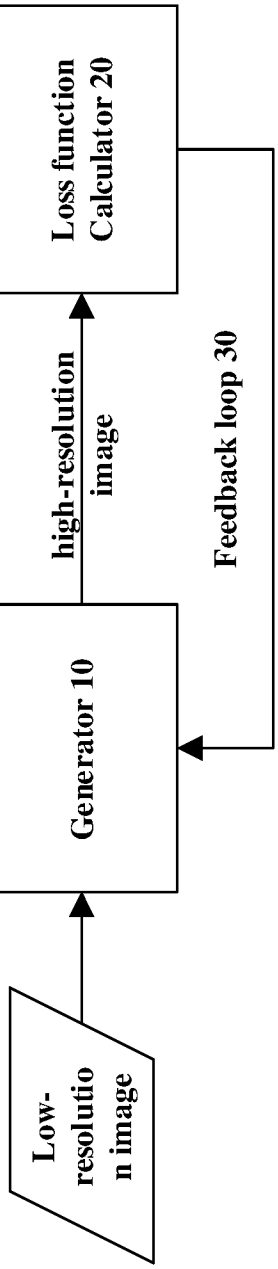

Low-resolution image
Generator 10
high-resolution image
Loss function Calculator 20
Feedback loop 30

FIG. 2

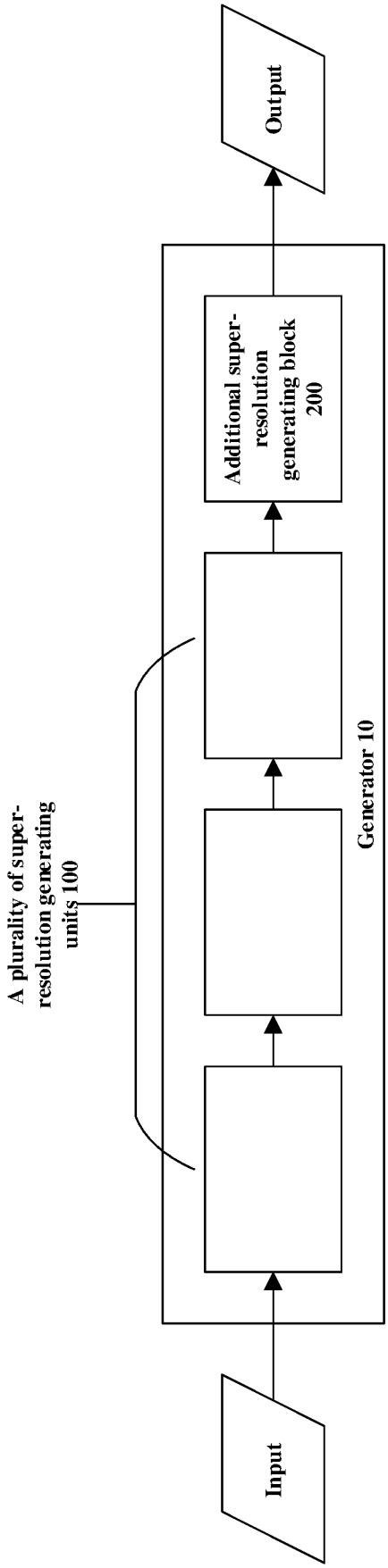

Input
Generator 10
A plurality of super-resolution generating units 100
Additional super-resolution generating block 200
Output

FIG. 3

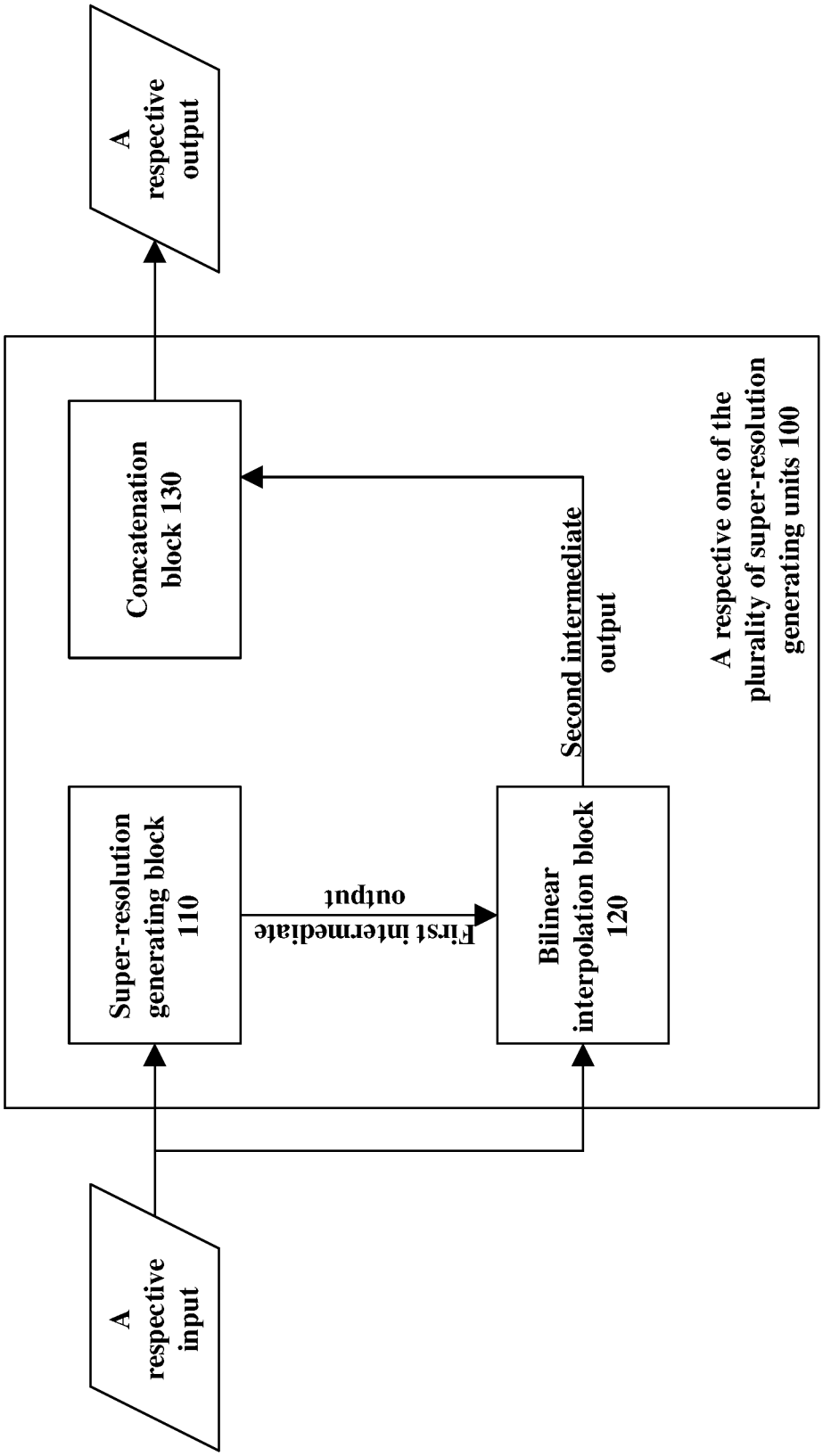

A respective input
Super-resolution generating block 110
First intermediate output
Bilinear interpolation block 120
Second intermediate output
Concatenation block 130
A respective output
A respective one of the plurality of super-resolution generating units 100

FIG. 4

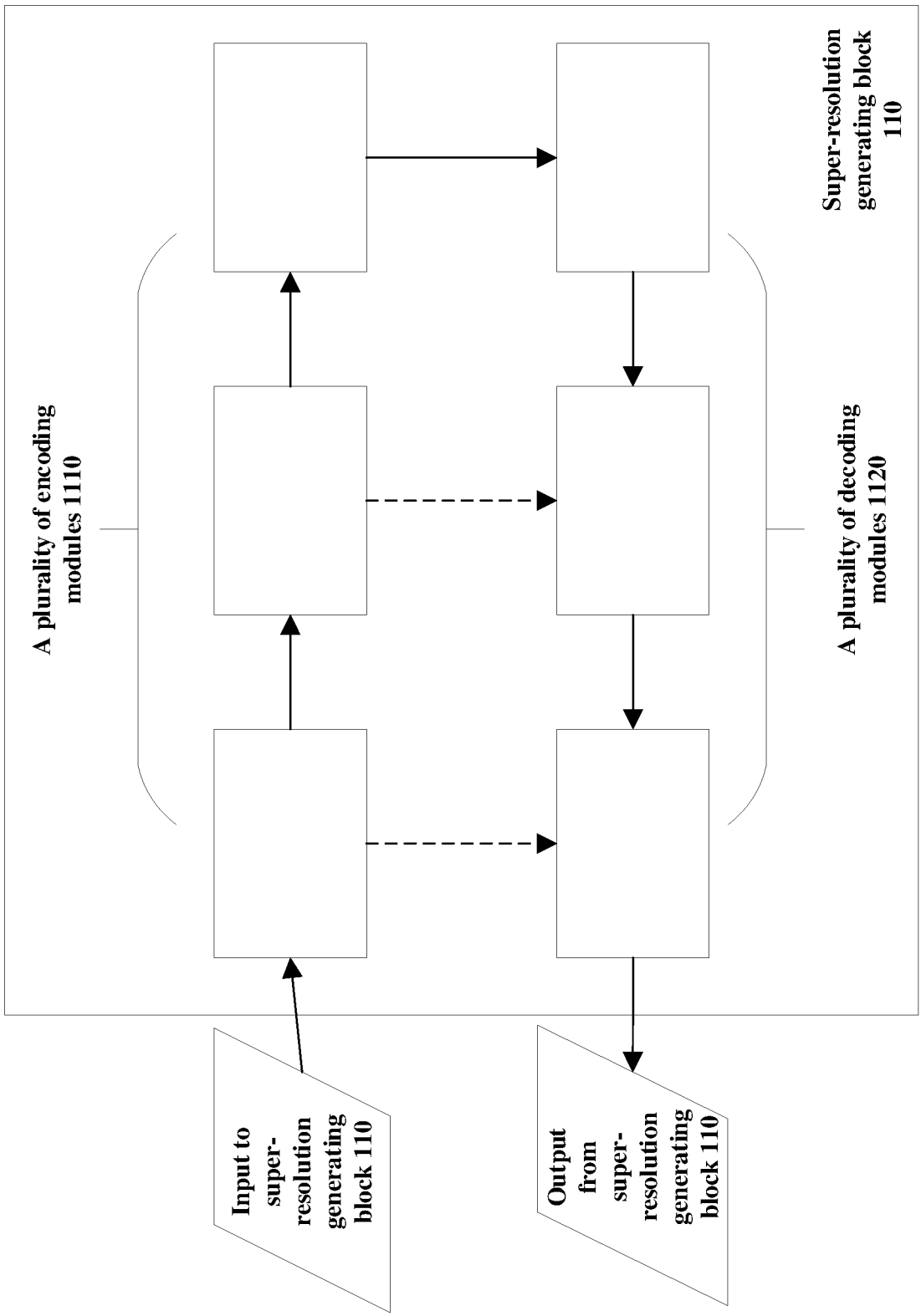

Super-resolution generating block 110
A plurality of encoding modules 1110
A plurality of decoding modules 1120
Input to super-resolution generating block 110
Output from super-resolution generating block 110

FIG. 5

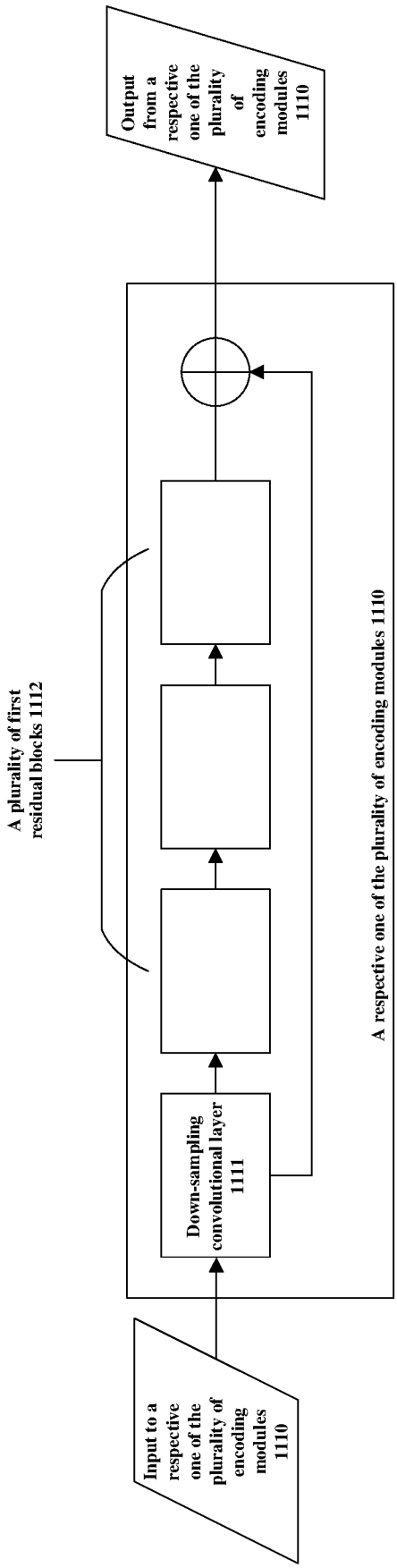

Input to a respective one of the plurality of encoding modules 1110
Down-sampling convolutional layer 1111
A plurality of first residual blocks 1112
Output from a respective one of the plurality of encoding modules 1110
A respective one of the plurality of encoding modules 1110

FIG. 6

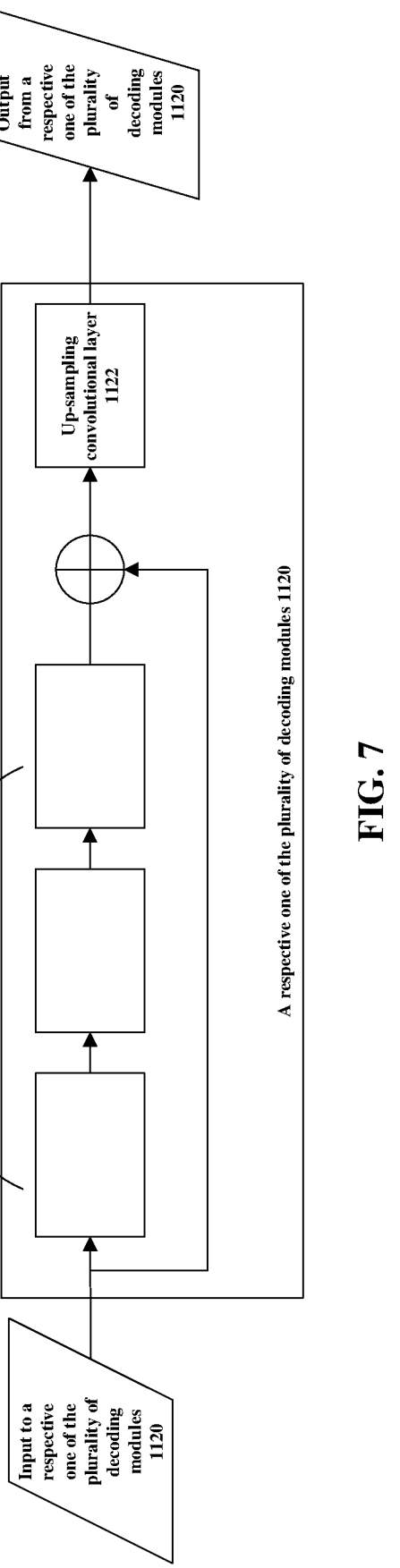

Input to a respective one of the plurality of decoding modules 1120
A plurality of first residual blocks 1121
Up-sampling convolutional layer 1122
Output from a respective one of the plurality of decoding modules 1120
A respective one of the plurality of decoding modules 1120

FIG. 7



FIG. 8A



FIG. 8B

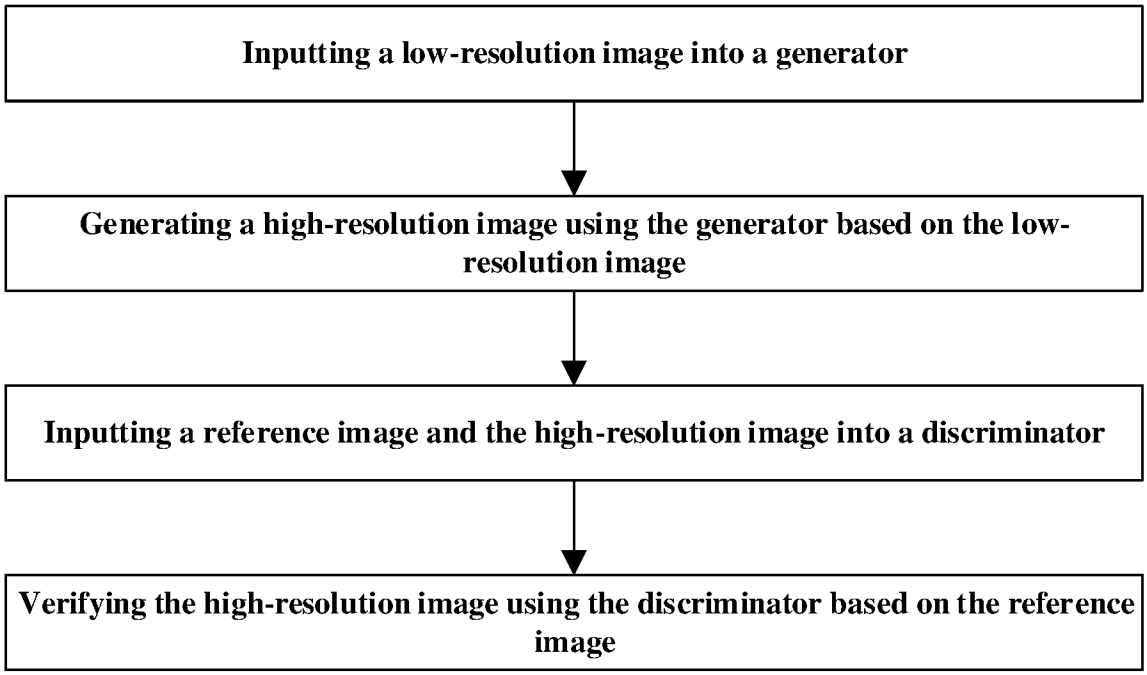

Inputting a low-resolution image into a generator
Generating a high-resolution image using the generator based on the low-resolution image
Inputting a reference image and the high-resolution image into a discriminator
Verifying the high-resolution image using the discriminator based on the reference image

FIG. 9

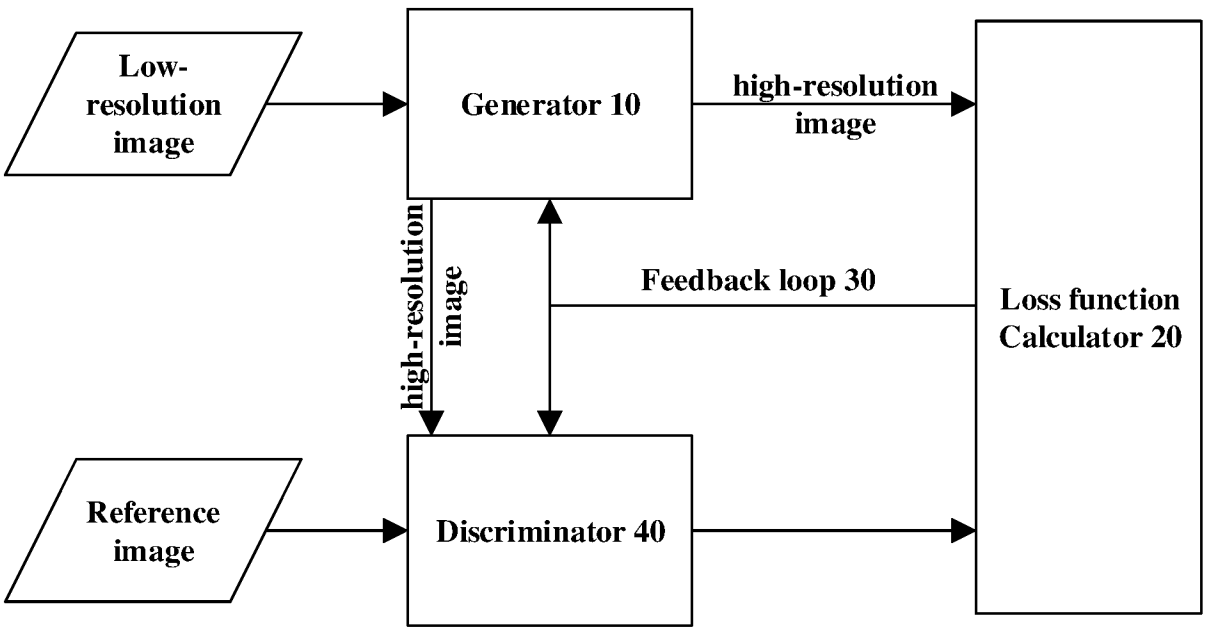

Low-resolution image
Generator 10
high-resolution image
high-resolution image
Feedback loop 30
Loss function Calculator 20
Reference image
Discriminator 40

FIG. 10

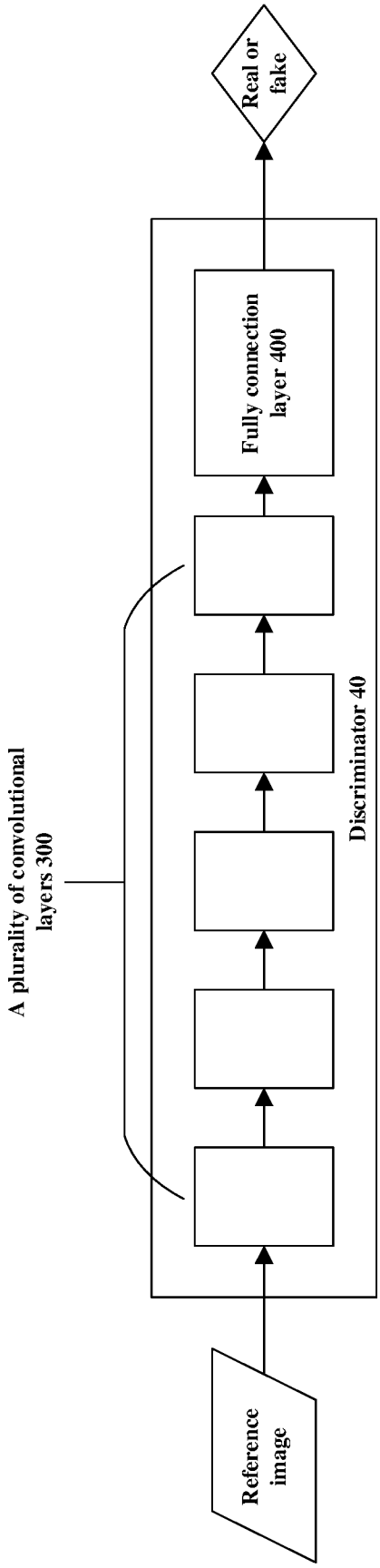

Real or fake
Fully connection layer 400
A plurality of convolutional layers 300
Discriminator 40
Reference image

FIG. 11



FIG. 12C



FIG. 12B



FIG. 12A



FIG. 13C

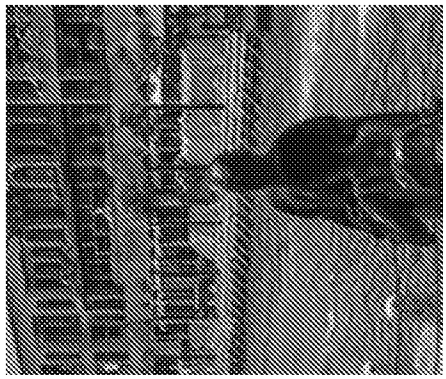


FIG. 13B

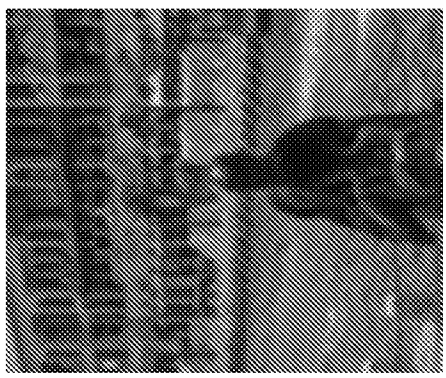


FIG. 13A

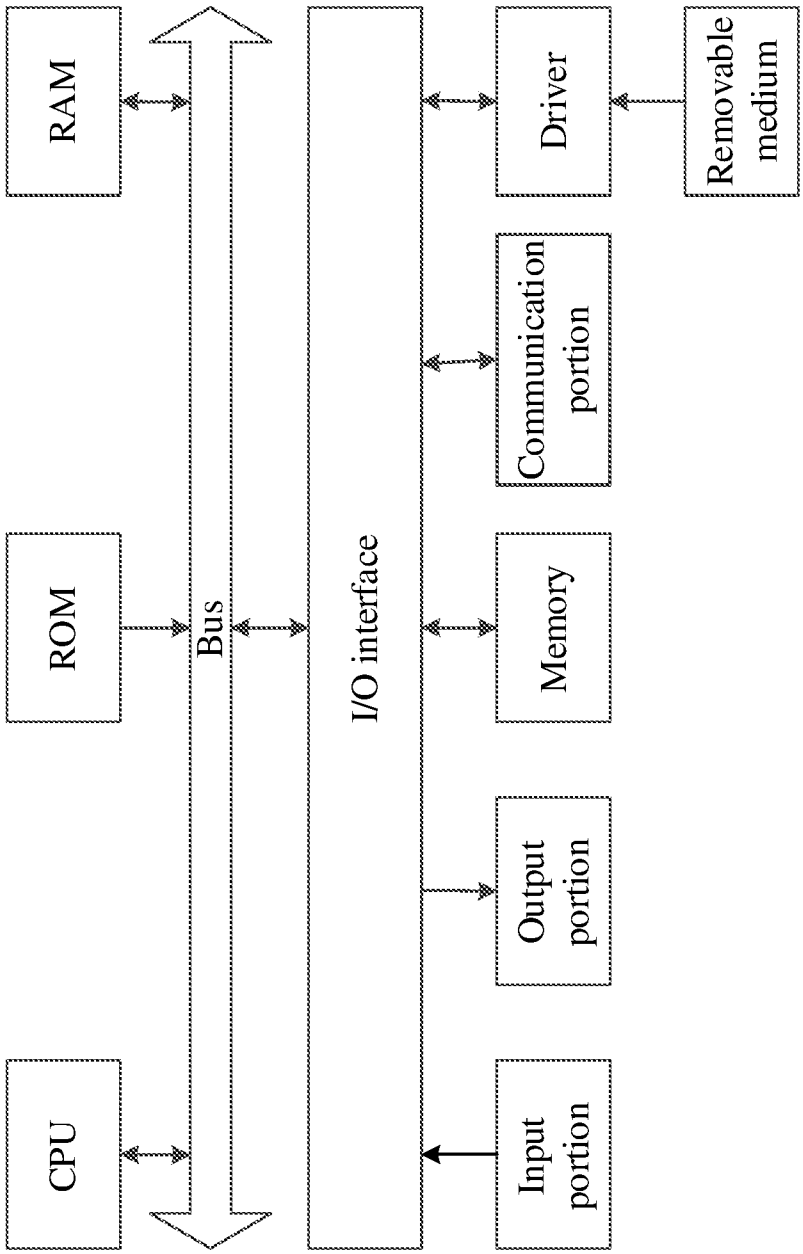

RAM
ROM
Bus
CPU
I/O interface
Input portion
Output portion
Memory
Communication portion
Driver
Removable medium

FIG. 14

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/082722

A. CLASSIFICATION OF SUBJECT MATTER

G06T 3/40(2006.01)i; G06T 5/00(2006.01)i; G06N 3/02(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06T,G06N,G06K

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

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

CNKI;CNPAT;WPI;EPODOC:series, plurality, neural, loss, network?, discriminator, resolution, concatenat+, multi+, dual, super, feedback, bilinear, three, image, interpolation, machine, picture, two, learn+, increas+, high, low, train

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2019259136 A1 (INTEL CORP.) 22 August 2019 (2019-08-22) description, paragraphs [0026]-[0143], and figures 1, 6, 10-15	1, 19-20
X	CN 109949255 A (HUAWEI TECHNOLOGIES CO., LTD.) 28 June 2019 (2019-06-28) description, paragraphs [0085]-[0173], and figures 1-6	1, 19-20
X	CN 109949224 A (BEIJING YUETU REMOTE SENSING TECHNOLOGY CO., LTD.) 28 June 2019 (2019-06-28) description, paragraphs [0028]-[0095], and figures 1-5	1, 19-20
A	CN 110298791 A (XI'AN UNIVERSITY OF POSTS AND TELECOMMUNICATIONS) 01 October 2019 (2019-10-01) the whole document	1-20
A	US 9367897 B1 (SHARP LAB. OF AMERICA, INC.) 14 June 2016 (2016-06-14) the whole document	1-20
A	JP 2014044497 A (JVC KENWOOD CORP.) 13 March 2014 (2014-03-13) the whole document	1-20

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

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

Date of the actual completion of the international search	Date of mailing of the international search report
11 December 2020	**30 December 2020**
Name and mailing address of the ISA/CN **National Intellectual Property Administration, PRC** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing** **100088** **China**	Authorized officer **HU,Ruixian**
Facsimile No. **(86-10)62019451**	Telephone No. **86-10-53961749**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/082722

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2019259136 A1	22 August 2019	None	
CN 109949255 A	28 June 2019	WO 2019120110 A1	27 June 2019
CN 109949224 A	28 June 2019	None	
CN 110298791 A	01 October 2019	None	
US 9367897 B1	14 June 2016	US 2016253784 A1	01 September 2016
		US 2016171656 A1	16 June 2016
JP 2014044497 A	13 March 2014	None	