

MODEL TRAINING METHOD AND APPARATUS, FEATURE EXTRACTION METHOD AND APPARATUS, DEVICE, AND MEDIUM

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority of Chinese Patent Application No. 202311518180.8, entitled “model training method, feature extraction method, apparatus, device, and medium”, filed with the China National Intellectual Property Administration on November 14, 2023, the entire content of which is incorporated herein by reference.

FIELD OF THE DISCLOSURE

[0002] The present disclosure generally relates to the technical field of data processing and, in particular, relates to a feature extraction model training method, a convolutional neural network training method, a model training method, a feature extraction method, an apparatus, a device, and a medium.

BACKGROUND

[0003] Secure multi-party computation (MPC) in privacy computation involves multiple parties collaborating to achieve computational goals without a trusted third party; and each party is guaranteed to receive no input information from other parties except for computation results, which effectively protects privacy information of users.

[0004] Herein, if the input information is an image containing private information of a user, to protect the private information of the user, feature extraction may be performed on the image, and secure multi-party computation is performed based on extracted feature vectors.

However, how to quickly and accurately extract high-precision image feature vectors is an urgent technical problem that needs to be solved.

BRIEF SUMMARY OF THE DISCLOSURE

[0005] The present application provides a feature extraction model training method, a convolutional neural network training method, a model training method, a feature extraction method, an apparatus, a device, and a medium for quickly and accurately extracting high-precision image feature vectors.

[0006] The first aspect of the present application provides a feature extraction model training method. The method includes:

[0007] for any obtained first sample image, obtaining a first feature vector of a first sample based on a trained convolutional neural network; processing the first sample image into a one-dimensional sequence based on a preset image processing manner, inputting the one-dimensional sequence into a feature extraction model to-be-trained, and obtaining a second feature vector of the first sample image based on the feature extraction model; and based on the second feature vector, calibrating the first feature vector to obtain a calibrated first baseline feature vector, where the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and

[0008] training the feature extraction model based on the first baseline feature vector, a label feature vector carried by the first sample image, and a configured first loss function.

[0009] In an optional implementation manner, after obtaining the second feature vector of the first sample image, and before calibrating the first feature vector based on the second feature vector, the method further includes:

[0010] performing global average pooling on the second feature vector to obtain global feature vector information corresponding to each dimension feature of the second feature vector, and obtaining an importance weight coefficient corresponding to each dimension feature of the global feature vector information based on a probability distribution function; based on the importance weight coefficient corresponding to each dimension feature, calibrating the second feature vector to obtain a calibrated second feature vector; and based on the calibrated second feature vector, performing a subsequent step of calibrating the first feature vector based on the second feature vector.

[0011] In an optional implementation manner, the first loss function includes the importance weight coefficient corresponding to each dimension feature.

[0012] In an optional implementation manner, calibrating the first feature vector based on the second feature vector to obtain the calibrated first baseline feature vector includes:

[0013] performing an XOR operation on each dimension feature of the second feature vector and each dimension feature of the first feature vector and obtaining the calibrated first baseline feature vector.

[0014] In an optional implementation manner, after performing the XOR operation on each dimension feature of the second feature vector and each dimension feature of the first feature vector and before obtaining the calibrated first baseline feature vector, the method further includes:

[0015] performing a normalization processing on vectors obtained after the XOR operation; and based on normalized vectors, determining the calibrated first baseline feature vector.

[0016] In an optional implementation manner, activation functions used in the feature extraction model includes a 2quad activation function.

[0017] In an optional implementation manner, the feature extraction model is a model obtained based on knowledge distillation.

[0018] The second aspect of the present application provides a convolutional neural network training method. The method includes:

[0019] for any obtained second sample image, inputting a second sample image into a convolutional neural network to-be-trained, and based on the convolutional neural network, obtaining a third feature vector of the second sample image; processing the second sample image into a one-dimensional sequence based on a preset image processing manner, inputting the one-dimensional sequence into a trained feature extraction model, and obtaining a fourth feature vector of the second sample image based on the feature extraction model; and based on the fourth feature vector, calibrating the third feature vector to obtain a calibrated second baseline feature vector, where the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and

[0020] training the convolutional neural network based on the second baseline feature vector, a label feature vector carried by the second sample image, and a configured second loss function.

[0021] In an optional implementation manner, based on the fourth feature vector, calibrating the third feature vector to obtain the calibrated second baseline feature vector includes:

[0022] performing an XOR operation on each dimension feature of the fourth feature vector and each dimension feature of the third feature vector and obtaining the calibrated second baseline feature vector.

[0023] In an optional implementation manner, after performing the XOR operation on each dimension feature of the fourth feature vector and each dimension feature of the third feature vector and before obtaining the calibrated second baseline feature vector, the method further includes:

[0024] performing a normalization processing on vectors obtained after the XOR operation; and based on normalized vectors, determining the calibrated second baseline feature vector.

[0025] The third aspect of the present application provides a model training method. The method includes:

[0026] for any obtained third sample image, obtaining a fifth feature vector of a third sample image based on a convolutional neural network to-be-trained; processing the third sample image into a one-dimensional sequence based on a preset image processing manner, inputting the one-dimensional sequence into a feature extraction model to-be-trained, and obtaining a sixth feature vector of the third sample image based on the feature extraction model; and based on the sixth feature vector, calibrating the fifth feature vector to obtain a calibrated third baseline feature vector, where the feature extraction model is a model trained based on a framework

structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and

[0027] training the feature extraction model based on the third baseline feature vector, a label feature vector carried by the third sample image, and a configured first loss function; and training the convolutional neural network based on the third baseline feature vector, the label feature vector carried by the third sample image, and a configured second loss function.

[0028] The fourth aspect of the present application provides a feature extraction method. The method includes:

[0029] receiving an image to-be-processed; and

[0030] obtaining a feature vector of the image to-be-processed based on the feature extraction model trained by any method in the first aspect and the third aspect or based on the convolutional neural network trained by any method in the second aspect and the third aspect.

[0031] In an optional implementation manner, the method further includes:

[0032] performing secure multi-party computation based on the feature vector.

[0033] The fifth aspect of the present application provides a feature extraction model training apparatus. The apparatus includes:

[0034] a first calibrating module, configured to, for any obtained first sample image, obtain a first feature vector of a first sample based on a trained convolutional neural network; process the first sample image into a one-dimensional sequence based on a preset image processing manner, input the one-dimensional sequence into a feature extraction model to-be-trained, and obtain a second feature vector of the first sample image based on the feature extraction model; and based on the second feature vector, calibrate the first feature vector to

obtain a calibrated first baseline feature vector, where the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and

[0035] a first training module, configured to train the feature extraction model based on the first baseline feature vector, a label feature vector carried by the first sample image, and a configured first loss function.

[0036] In an optional implementation manner, the first calibrating module is further configured to:

[0037] perform global average pooling on the second feature vector to obtain global feature vector information corresponding to each dimension feature of the second feature vector, and obtain an importance weight coefficient corresponding to each dimension feature of the global feature vector information based on a probability distribution function; based on the importance weight coefficient corresponding to each dimension feature, calibrate the second feature vector to obtain a calibrated second feature vector; and based on the calibrated second feature vector, perform a subsequent step of calibrating the first feature vector based on the second feature vector.

[0038] In an optional implementation manner, the first loss function includes the importance weight coefficient corresponding to each dimension feature.

[0039] In an optional implementation manner, the first calibrating module is specifically configured to:

[0040] perform an XOR operation on each dimension feature of the second feature vector and each dimension feature of the first feature vector and obtain the calibrated first baseline feature vector.

[0041] In an optional implementation manner, the first calibrating module is further configured to:

[0042] perform a normalization processing on vectors obtained after the XOR operation; and based on normalized vectors, determine the calibrated first baseline feature vector.

[0043] In an optional implementation manner, activation functions used in the feature extraction model includes a 2quad activation function.

[0044] In an optional implementation manner, the feature extraction model is a model obtained based on knowledge distillation.

[0045] The sixth aspect of the present application provides a convolutional neural network training apparatus. The apparatus includes:

[0046] a second calibrating module, configured to, for any obtained second sample image, input a second sample image into a convolutional neural network to-be-trained, and obtain a third feature vector of the second sample image based on the convolutional neural network; process the second sample image into a one-dimensional sequence based on a preset image processing manner, input the one-dimensional sequence into a trained feature extraction model, and obtain a fourth feature vector of the second sample image based on the feature extraction model; and based on the fourth feature vector, calibrate the third feature vector to obtain a calibrated second baseline feature vector, where the feature extraction model is a model

trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and

[0047] a second training module, configured to train the convolutional neural network based on the second baseline feature vector, a label feature vector carried by the second sample image, and a configured second loss function.

[0048] In an optional implementation manner, the second calibrating module is specifically configured to:

[0049] perform an XOR operation on each dimension feature of the fourth feature vector and each dimension feature of the third feature vector and obtain the calibrated second baseline feature vector.

[0050] In an optional implementation manner, the second calibrating module is further configured to:

[0051] perform a normalization processing on vectors obtained after the XOR operation; and based on normalized vectors, determine the calibrated second baseline feature vector.

[0052] The seventh aspect of the present application provides a model training apparatus. The apparatus includes:

[0053] a third calibrating module, configured to, for any obtained third sample image, obtain a fifth feature vector of a third sample image based on a convolutional neural network to-be-trained; process the third sample image into a one-dimensional sequence based on a preset image processing manner, input the one-dimensional sequence into a feature extraction model to-be-trained, and obtain a sixth feature vector of the third sample image based on the feature extraction model; and based on the sixth feature vector, calibrate the fifth feature vector to obtain

a calibrated third baseline feature vector, where the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and

[0054] a third training module, configured to train the feature extraction model based on the third baseline feature vector, a label feature vector carried by the third sample image, and a configured first loss function; and train the convolutional neural network based on the third baseline feature vector, the label feature vector carried by the third sample image, and a configured second loss function.

[0055] The eighth aspect of the present application provides a feature extraction apparatus. The apparatus includes:

[0056] a receiving module, configured to receive an image to-be-processed; and

[0057] an obtaining module, configured to obtain a feature vector of the image to-be-processed based on the feature extraction model trained by any method in the first aspect and the third aspect, or the convolutional neural network trained by any method in the second aspect and the third aspect.

[0058] In an optional implementation manner, the apparatus further includes:

[0059] a secure multi-party computation module, configured to perform secure multi-party computation based on the feature vector.

[0060] The ninth aspect of the present application provides an electronic device, where the electronic device at least includes a processor and a memory, and when executing a computer program stored in the memory, the processor is configured to implement steps of any method mentioned above.

[0061] The tenth aspect of the present application provides a computer-readable storage medium, storing a computer program, where when executed by a processor, the computer program implements steps of any method mentioned above.

[0062] In embodiments of the present application, the first sample image may be processed into the one-dimensional sequence, and the one-dimensional sequence is inputted into the feature extraction model, such that the feature extraction model may extract features from the first sample image; furthermore, the feature extraction model in embodiments of the present application is a model trained using the framework structure of the MPCFormer model and the training logic of the SETR model, such that the feature extraction model may capture more accurate facial feature vectors (the second feature vector) based on the self-attention mechanism, and calibrate the first feature vector obtained by the convolutional neural network based on the second feature vector to obtain the first baseline feature vector that more accurately represents image features; and the feature extraction model is trained based on the first baseline feature vector and configured first loss function, thereby improving the accuracy of image feature vectors extracted by the trained feature extraction model and implementing fast and accurate extraction of high-precision image feature vectors.

[0063] Furthermore, the feature extraction model in embodiments of the present application is a model trained using the framework structure of the MPCFormer model and the training logic of the SETR model, such that the feature extraction model may obtain pixel-level accurate feature vectors, thereby further improving the accuracy of image feature vectors determined (or extracted) by the trained feature extraction model. Moreover, in embodiments of the present application, the self-attention mechanism may be configured to weaken or remove (ignore) unimportant features in the first feature vector and the like, thereby obtaining fewer and

more precise feature vectors, improving the accuracy of obtained feature vectors, reducing computation time and increasing efficiency while also mitigating the communication bottleneck of MPC; and additional computational costs may be not needed, thereby reducing the communication bottleneck of MPC.

BRIEF DESCRIPTION OF THE DRAWINGS

[0064] To more clearly illustrate the implementation manners in embodiments of the present application or related technology, accompanying drawings used in the description of the embodiments or related technology are briefly described below. Obviously, accompanying drawings in the following description are merely certain embodiments of the present application. For those skilled in the art, other accompanying drawings may be also obtained from such accompanying drawings without any creative effort.

[0065] FIG. 1 illustrates a schematic of a training process of a first feature extraction model provided in some embodiments of the present application.

[0066] FIG. 2A illustrates an image schematic corresponding to a first feature vector obtained based on a convolutional neural network in existing technology provided in embodiments of the present application.

[0067] FIG. 2B illustrates an image schematic corresponding to a second feature vector obtained based on a feature extraction model provided in embodiments of the present application.

[0068] FIG. 3 illustrates a schematic of a training process of a second feature extraction model provided in embodiments of the present application.

[0069] FIG. 4 illustrates a schematic of a training process of a third feature extraction model provided in embodiments of the present application.

[0070] FIG. 5 illustrates a diagram of communication durations of different activation functions provided in embodiments of the present application.

[0071] FIG. 6 illustrates a schematic of a Transformer model provided in embodiments of the present application.

[0072] FIG. 7 illustrates a schematic of a SETR model provided in embodiments of the present application.

[0073] FIG. 8 illustrates a schematic of a convolutional neural network training process provided in embodiments of the present application.

[0074] FIG. 9 illustrates a schematic of a secure multi-party computation process provided in embodiments of the present application.

[0075] FIG. 10 illustrates a schematic of a model training process provided in embodiments of the present application.

[0076] FIG. 11 illustrates a schematic a feature extraction process provided in embodiments of the present application.

[0077] FIG. 12 illustrates a schematic of a feature extraction model training apparatus provided in embodiments of the present application.

[0078] FIG. 13 illustrate a schematic of a convolutional neural network training apparatus provided in embodiments of the present application.

[0079] FIG. 14 illustrate a schematic of a model training apparatus provided in embodiments of the present application.

[0080] FIG. 15 illustrate a schematic of a feature extraction apparatus provided in embodiments of the present application.

[0081] FIG. 16 illustrate a structural schematic of an electronic device provided in embodiments of the present application.

DETAILED DESCRIPTION

[0082] To quickly and accurately extract (obtain) high-precision image feature vectors, the present application provides a feature extraction model training method, a convolutional neural network training method, a model training method, a feature extraction method, an apparatus, a device, and a medium.

[0083] To make the objectives and implementation manners of the present disclosure clearer, embodiments of the present disclosure are clearly and completely described below with reference to accompanying drawings in embodiments. Obviously, described embodiments are only a part, not all, of embodiments of the present disclosure.

[0084] It should be noted that brief description of terms in the present disclosure may be only for the convenience of understanding embodiments described below and may be not intended to limit the implementation manners of the present disclosure. Unless otherwise stated, such terms may be understood in ordinary and common meanings.

[0085] The terms “first”, “second”, “third”, and the like in the specification, claims, and accompanying drawings of the present disclosure may be configured to distinguish similar or

related objects or entities and may not necessarily imply a specific order or sequence. Unless otherwise specified, it may be understood that such terms may be used interchangeably in appropriate contexts.

[0086] The terms “comprise” and “have”, and any variations thereof may be intended to be omniscient but not exclusive. For example, a product or device including a series of components may be not necessarily limited to all components explicitly listed but may include other components not explicitly listed or inherent to such product or device.

[0087] The term “module” may refer to any known or subsequently developed hardware, software, firmware, artificial intelligence, fuzzy logic, or the combination of hardware and/or software code, which may be capable of performing functions associated with such component.

[0088] Finally, it should be noted that above-mentioned embodiments may merely describe the technical solutions of the present disclosure and may be not intended to limit the technical solutions. Although the present disclosure has been described in detail with reference to above embodiments, those skilled in the art should understand that modifications may still be made to the technical solutions described in above embodiments, or equivalent substitutions may be made to some or all technical features. In addition, such modifications or substitutions may not cause the essence of corresponding technical solutions to deviate from the scope of the technical solutions in embodiments of the present disclosure.

[0089] All implementation manners of embodiments of the present disclosure comply with relevant provisions of national laws and regulations regarding data acquisition, storage, use, processing and the like.

[0090] Embodiment one

[0091] FIG. 1 illustrates a schematic of a training process of a first feature extraction model provided in some embodiments of the present application. As shown in FIG. 1, such process includes following steps.

[0092] At S101, for any obtained first sample image, the first feature vector of the first sample is obtained based on a trained convolutional neural network; the first sample image is processed into a one-dimensional sequence based on a preset image processing manner, the one-dimensional sequence is inputted into a feature extraction model to-be-trained, and the second feature vector of the first sample image is obtained based on the feature extraction model; and based on the second feature vector, the first feature vector is calibrated to obtain a calibrated first baseline feature vector, where the feature extraction model is a model trained based on a framework structure of a MPCFormer (multi-party computation transformer) model and a training logic of a semantic vision transformer model SETR (Segmentation Transformer).

[0093] In a possible implementation manner, the feature extraction model may a model trained based on the framework structure of a MPCFormer model and the training logic of a semantic vision transformer model (SETR), where how the feature extraction model in embodiments of the present application is obtained based on the framework structure of the MPCFormer model and the training logic of the SETR model is described subsequently, which may be not described in detail herein.

[0094] In a possible implementation manner, during training of the feature extraction model, in order to improve the accuracy of the feature vectors obtained by the feature extraction model during training, any first sample image may be obtained from the first sample image set; the first sample image obtained is inputted into a trained convolutional neural network (CNN), such as a deep residual network (ResNet) and the like; and the convolutional neural network

performs feature extraction on the first sample image to obtain the first feature vector. Herein, the first sample image may contain user privacy information, such as biometric features like facial feature and the like, and the present application does not specifically limit the type of the first sample image.

[0095] In a possible implementation manner, the first sample image may be further processed into a one-dimensional sequence based on a preset image processing, the one-dimensional sequence is inputted into the feature extraction model to-be-trained, and feature extraction is performed on the first sample image based on the feature extraction model to obtain the second feature vector; and based on the second feature vector, the first feature vector is calibrated to obtain a more accurate feature vector (the first baseline feature vector).

[0096] Specifically, considering that the framework structure of the feature extraction model is a framework structure based on the MPCFormer model and the MPCFormer model in related technology is typically used for natural language processing, in order to enable the feature extraction model to process images, in embodiments of the present application, the first sample image may be processed into the one-dimensional sequence, the one-dimensional sequence is inputted into the feature extraction model, and the training logic for the semantic vision transformer (SEgmentation Transformer SETR) model is configured to train the feature extraction model, such that that the operational logic of the feature extraction model is same as the operational logic of the SETR model. Therefore, the feature extraction model may not only process images but also obtain more precise image features at the pixel level based on the SETR model's operational logic; moreover, based on multi-head self-attention mechanism and the like, the feature extraction model may obtain more accurate (few but precise) image features (feature vectors), thereby improving the speed and accuracy of secure multi-party computation (MPC).

Herein, how the first sample image is processed into the one-dimensional sequence and how the feature extraction model in embodiments of the present application is obtained based on the framework structure of the MPCFormer model and the training logic of the SETR model are described subsequently; and the process of calibrating the first feature vector based on the second feature vector is first explained below.

[0097] For ease of understanding, in embodiments of the present application, F is used to denote the first feature vector, and A is used to denote the second feature vector; and when calibrating the first feature vector based on the second feature vector A , the first feature vector F may be pooled using the second feature vector A , that is, XOR operation ($\odot$) is performed on each dimension feature of the second feature vector (attention maps) and each dimension feature of the first feature vector. For example, the vector obtained after performing XOR operation on the k -th vector (attention map) in the second feature vector (attention maps) is $v_k = \sum_{i,j} F_{i,j,k} \odot A_{i,j,k}$ where “ i,j,k ” denotes the pixel in the i -th row and j -th column of the k -th vector (attention map).

[0098] In a possible implementation manner, when performing XOR operation on each dimension feature included in the second feature vector and each dimension feature in the first feature vector, when the pixel values of corresponding feature in the second feature vector and the first feature vector are both relatively high, the feature is considered to be relatively desirable, and the pixel value of corresponding feature obtained after XOR operation is set to relatively high. However, when the pixel value of corresponding feature in either the first feature vector or the second feature vector is relatively low, the feature is considered to be relatively poor, and the pixel value of corresponding feature obtained after the XOR operation is set to relatively low,

thereby obtaining more accurate (few but precise) image features (feature vectors) through calibration.

[0099] Optionally, taking the second feature vector as an example, if the pixel value of a feature is higher than the average pixel value of the features in the second feature vector, the pixel value of such feature may be considered to be relatively high; conversely, if the pixel value of a feature is not higher than the average pixel value of the features in the second feature vector, the pixel value of such feature may be considered to be relatively low. Similarly, if the pixel value of a feature in the first feature vector is higher than the average pixel value of the features in the first feature vector, the pixel of such feature may be considered to be relatively high; conversely, if the pixel value of such feature is not higher than the average pixel value of the features in the first feature vector, the pixel value of such feature may be considered to be relatively low. Herein, specific computation process described above for performing XOR operation on each dimension feature of the second feature vector and each dimension feature of the first feature vector is merely exemplary, and the present application does not specifically limit specific computation process of the XOR operation.

[00100] In a possible implementation manner, after performing XOR operation on each dimension feature of the second feature vector and each dimension feature of the first feature vector, based on fully connected layer, normalization processing is performed on the vectors obtained after XOR operation, thereby being normalized to the feature space of same size. Next, normalized vectors are averaged to obtain the calibrated first baseline feature vector, i.e., final calibrated first baseline feature vector $f = \frac{1}{K} \sum_k V_k$.

[00101] Referring to FIGS. 2A-2B, FIG. 2A illustrates an image schematic corresponding to a first feature vector obtained based on a convolutional neural network in existing technology

provided in embodiments of the present application; and FIG. 2B illustrates an image schematic corresponding to a second feature vector obtained based on a feature extraction model provided in embodiments of the present application. Herein, in FIG. 2B, the black region is irrelevant feature region. It may be seen that the feature extraction model may utilize the self-attention mechanism to capture more specific facial features in the image while ignoring other irrelevant feature regions in the image; and the attention mechanism enables features of image major regions having high information value to be extracted. Specifically, for facial features, the effect may be that attention may filter out facial features while masking other image information such as ambient lighting, background and the like, and represent such other image information with black region, thereby obtaining fewer but more precise and accurate facial features.

[00102] In a possible implementation manner, taking the first sample image being an image including a face as an example, the face in the image may have occluded or tilted portions. To suppress features of such portions and obtain more prominent and accurate facial features, after obtaining the second feature vector A of the first sample image and before calibrating the first feature vector F based on the second feature vector A , the second feature vector A may be calibrated; and the first feature vector is calibrated based on the calibrated second feature vector $\tilde{A}$, thereby further improving the accuracy of obtained first baseline feature vector. Herein, the process of calibrating the second feature vector A to obtain the calibrated second feature vector $\tilde{A}$ can be described as the following.

[00103] Optionally, global average pooling (GAP) may be performed on the second feature vector A firstly to obtain global feature vector information corresponding to each dimension feature of the second feature vector; obtained global feature vector information may be mapped to the softmax probability distribution function ($f_{\text{ex}}(\cdot) = \text{softmax}(\cdot)$); based on such

softmax probability distribution function (also known as normalized exponential function), relatively important feature information in the second feature vector A is further amplified, and the importance weight coefficient s corresponding to each dimension feature in the global feature vector information is obtained. Exemplarily, the importance weight coefficient s may be:

$f_{\text{ex}}(\frac{1}{20^2} \sum_{i,j} A_{i,j,k})$. Herein, the more important features in the second feature vector (such as un-obstructed and un-tilted facial features and the like) have larger importance weight coefficients s , while the less important features (such as occluded or tilted facial features and the like) have smaller importance weight coefficients s , thereby suppressing facial features of the portions that are occluded or titled; and lower activation levels (weight coefficients) are assigned to regions in the attention maps corresponding to occluded or tilted portions, thereby further improving the accuracy of determined first baseline feature vector.

[00104] In a possible implementation manner, the second feature vector A may be transformed using the sigmoid function; and the transformed second feature vector (e.g., $\text{sigmoid}(A_k)$) is then multiplied by corresponding importance weight coefficient s_k to obtain the calibrated second feature vector $A^{\sim}$. Exemplarily, in the calibrated second feature vector, $A_k^{\sim} = s_k \cdot \text{sigmoid}(A_k)$.

[00105] After obtaining the calibrated second feature vector $A^{\sim}$, the first feature vector F may be calibrated based on the calibrated second feature vector $A^{\sim}$ to obtain the first baseline feature vector f . Herein, the process of calibrating the first feature vector F based on the calibrated second feature vector $A^{\sim}$ is similar to the process of calibrating the first feature vector F based on the second feature vector A . For example, XOR operation ($\odot$) is performed on each dimension feature of the second feature vector (attention maps) $A^{\sim}$ and each dimension feature of

the first feature vector, and the k-th vector (attention map): $v_k = \sum_{i,j} F_{i,j,k} \odot A_{i,j,k}$, which is not described in detail herein.

[00106] At S102, the feature extraction model is trained based on the first baseline feature vector, the label feature vector carried by the first sample image, and the configured first loss function.

[00107] In a possible implementation manner, the first sample image may carry a pre-configured label feature vector, where the configuration process of the label feature vector is not specifically limited in the present application. The feature extraction model may be trained based on the first baseline feature vector f , the label feature vector carried by the first sample image, and the configured first loss function L , thereby obtaining a feature extraction model capable of being deployed online for extracting image feature vectors (such as feature vectors of face images).

[00108] Optionally, the first loss function L may include a cross-entropy loss function, a weighted differential regularization loss function, and an L2 regularization loss function. To extract high-precision image feature vectors, the first loss function may include importance weight coefficients corresponding to each dimension feature. Exemplarily, when configuring the cross-entropy loss function in the first loss function L , the cross-entropy loss function may be determined based on the importance weight coefficients s_k mentioned above and the cross-entropy loss term parameter $L_{CE,k}$. For example, the cross-entropy loss function may be determined by summing the products of the importance weight coefficients s_k corresponding to each dimension feature and corresponding cross-entropy loss term parameter $L_{CE,k}$. For ease of understanding, the cross-entropy loss function L_{wCE} may be expressed as: $L_{wCE} = \sum_k s_k \cdot L_{CE,k}$. The cross-entropy loss function L_{wCE} in embodiments of the present application may be

configured to allow the loss caused by occluded or tilted portions of the attention maps to model convergence to be constrained by own importance weight coefficients s , such that the loss associated with such portions does not participate in model convergence computation, thereby improving the accuracy of feature vectors determined by the trained model (feature extraction model). Herein, the cross-entropy loss term parameter $L_{CE,k}$ may be determined based on existing technology, which may be not described in detail herein.

[00109] Optionally, when configuring weighted differential regularization loss function in the first loss function L , the weighted differential regularization loss function L_{wDIV} may be determined based on the importance weight coefficients s_k mentioned above and the weighted differential loss term parameter $P_{i,j,k}$. For ease of understanding, the weighted differential regularization loss function L_{wDIV} may be expressed as: $L_{wDIV} = 1 - \sum_{i,j} \max_k(s_k \cdot P_{i,j,k})$, where the weighted differential loss term parameter $P_{i,j,k} = \frac{\exp(A_{i,j,k})}{\sum_{i,j} \exp(A_{i,j,k})}$. The weighted differential loss term parameter $P_{i,j,k}$ may be considered as a penalty term for cross-overlapping features; and based on the weighted differential regularization loss function L_{wDIV} , features with cross-overlapping features in the feature vector may be weakened or removed, such that the loss and the like caused by such features does not participate in the calculation of model convergence, thereby improving the accuracy of the feature vector determined by the trained model (feature extraction model). Herein, the weighted differential loss term parameters $P_{i,j,k}$ may be determined using existing technology, which may be not described in detail herein. Furthermore, the L2 regularization loss function L_{REG} may also be determined using existing technology, which may be not described in detail herein.

[00110] Optionally, a corresponding weight may be configured for each of three sub-loss functions in the first loss function: the cross-entropy loss function L_{wCE} , the weighted differential regularization loss function L_{wDIV} , and the L2 regularization loss function L_{REG} ; and the loss function L is determined based on the sum of the products of each sub-loss function and corresponding weight. Exemplarily, the weight corresponding to the cross-entropy loss function L_{wCE} is denoted by λ_{wCE} , the weight corresponding to the weighted differential regularization loss function L_{wDIV} is denoted by λ_{wDIV} , and the weight corresponding to the L2 regularization loss function L_{REG} is denoted by λ_{REG} . Therefore, the first loss function $L = \lambda_{wCE}L_{wCE} + \lambda_{wDIV}L_{wDIV} + \lambda_{REG}L_{REG}$. Herein, specific training process for training the feature extraction model based on the first loss function may employ existing technology, which may be not described in detail herein. The present application does not specifically limit the weights corresponding to each sub-loss function which may be flexibly configured according to requirements.

[00111] Optionally, when training the feature extraction model, whether a recognition result is accurate may be determined by whether the first baseline feature vector is consistent with the label feature vector carried by the first sample image. In a specific implementation, if inconsistent, it indicates that the recognition result of the feature extraction model is not accurate, and the parameters of the feature extraction model need to be adjusted, such that the first baseline feature vector obtained based on the feature extraction model gradually approaches the label feature vector, thereby training the feature extraction model.

[00112] In a specific implementation, when adjusting the parameters in the feature extraction model, a gradient descent algorithm may be configured to backpropagate the gradient of the parameters of the feature extraction model, thereby training the feature extraction model.

[00113] In a possible implementation manner, above-mentioned operation may be performed on each first sample image in the first sample set; and when a preset convergence condition is satisfied, the training of the feature extraction model is considered complete. Herein, the preset convergence condition may be that the quantity of sample images correctly recognized from the first sample images in the first sample set through original feature extraction model is greater than a set quantity, or the quantity of iterations for training the feature extraction model reaches a set maximum quantity of iterations, which may be flexibly configured in specific implementations and may be not specifically limited herein.

[00114] In a possible implementation manner, when training original feature extraction model, the sample images in the sample set may be divided into training sample images and test sample images; and original feature extraction model is first trained based on the training sample images, and the reliability of trained feature extraction model is verified based on the test sample images.

[00115] In embodiments of the present application, the first sample image may be processed into the one-dimensional sequence, and the one-dimensional sequence is inputted into the feature extraction model, such that the feature extraction model may extract features from the first sample image; furthermore, the feature extraction model in embodiments of the present application is a model trained using the framework structure of the MPCFormer model and the training logic of the SETR model, such that the feature extraction model may capture more accurate facial feature vectors (the second feature vector) based on the self-attention mechanism, and calibrate the first feature vector obtained by the convolutional neural network based on the second feature vector to obtain the first baseline feature vector that more accurately represents image features; and the feature extraction model is trained based on the first baseline feature

vector and configured first loss function, thereby improving the accuracy of image feature vectors extracted by the trained feature extraction model and implementing fast and accurate extraction of high-precision image feature vectors.

[00116] Furthermore, the feature extraction model in embodiments of the present application is a model trained using the framework structure of the MPCFormer model and the training logic of the SETR model, such that the feature extraction model may obtain pixel-level accurate feature vectors, thereby further improving the accuracy of image feature vectors determined (or extracted) by the trained feature extraction model. Moreover, in embodiments of the present application, the self-attention mechanism may be configured to weaken or remove (ignore) unimportant features in the first feature vector and the like, thereby obtaining fewer and more precise feature vectors, improving the accuracy of obtained feature vectors, reducing computation time and increasing efficiency while also mitigating the communication bottleneck of MPC; and additional computational costs may be not needed, thereby reducing the communication bottleneck of MPC.

[00117] For ease of understanding, the training process of the feature extraction model provided in the present application is described through a specific embodiment hereinafter. Referring to FIG. 3, FIG. 3 illustrates a schematic of a training process of the second feature extraction model provided in embodiments of the present application. Such process includes following steps.

[00118] At S301, for any obtained first sample image, the first feature vector of the first sample is obtained based on trained convolutional neural network; the first sample image is processed into the one-dimensional sequence based on a preset image processing manner, the one-dimensional sequence is inputted into the feature extraction model to-be-trained; and the

second feature vector of the first sample image is obtained based on the feature extraction model, where the feature extraction model is a model trained based on the framework structure of the MPCFormer model and the training logic of the SETR model.

[00119] At S302, global average pooling is performed on the second feature vector to obtain global feature vector information corresponding to each dimension feature of the second feature vector, and an importance weight coefficient corresponding to each dimension feature of the global feature vector information is obtained based on a probability distribution function; based on the importance weight coefficient corresponding to each dimension feature, the second feature vector is calibrated to obtain a calibrated second feature vector; and based on the calibrated second feature vector, the first feature vector is calibrated to obtain a calibrated first baseline feature vector.

[00120] At S303, the feature extraction model is trained based on the first baseline feature vector, the label feature vector carried by the first sample image, and the configured first loss function.

[00121] For ease of understanding, the training process of the feature extraction model provided in the present application is further described through a specific embodiment hereinafter. Referring to FIG. 4, FIG. 4 illustrates a schematic of a training process of the third feature extraction model provided in embodiments of the present application. Such process includes following steps.

[00122] After obtaining any first sample image, the first sample image may be inputted into a trained convolutional neural network (such as a ResNet or other CNN neural networks); based on the convolutional neural network, feature extraction is performed on the first sample image to obtain the feature map of the first sample image, that is, the first feature vector F

(feature maps). Simultaneously, the first sample image may be processed into the one-dimensional sequence, where the process of processing the first sample image into the one-dimensional sequence may be as follows:

[00123] a 2D first sample image (such as a face image) with length $\times$ width $\times$ quantity of color channels (C) of $H \times W \times 3$ (unit: pixels) may be split into multiple small blocks of $\frac{H}{16} \times \frac{W}{16} = \frac{HW}{256}$; such small blocks are stretched into a one-dimensional (1D) sequence of length $L = HW/256$; and the one-dimensional sequence is configured as input embedding and inputted into the feature extraction model (Transformer) to-be-trained. Optionally, a positional embedding algorithm may be configured to encode the positional information of original image, and a corresponding positional code p_i is added to each vector e_i in the one-dimensional sequence, resulting in final one-dimensional sequence (input sequence) E inputted to the feature extraction model, where E may be represented as: $\{e_1+p_1, e_2+p_2, \dots, e_L+p_L\}$.

[00124] In a possible implementation manner, the one-dimensional sequence may be inputted into the feature extraction model to-be-trained, the framework structure of the feature extraction model may adopt the framework structure of the MPCFormer model, and the computational logic (also known as the training logic) may adopt the computational logic of the SETR model; and based on the encoder in the feature extraction model, the second feature vectors (attention maps) extracted using self-attention mechanism may be obtained. Optionally, each encoder in the feature extraction model may include a multi-head self-attention (MSA) layer and a feedforward neural network (also known as a multi-layer perceptron, MLP); and a total of L_e encoders is assumed. Herein, the input to the l -th self-attention layer is a three-dimensional tuple (query, key and value) calculated by Z^{l-1} , where query = $Z^{l-1} W_Q$, key = $Z^{l-1} W_K$, and value = $Z^{l-1} W_V$. Herein, W_Q , W_K , and W_V are learnable weight matrices, and d is

the dimension of the 3D tuple (query, key and value). Therefore, self-attention may be expressed as:

$$[00125] \quad SA(Z^{l-1}) = Z^{l-1} + \text{softmax}\left(\frac{Z^{l-1}W_Q(Z^{l-1}W_K)^T}{\sqrt{d}}\right)Z^{l-1}W_V.$$

[00126] Multi-head (multi-layer) self-attention is obtained by concatenating a number m of SAs, $MSA(Z^{l-1}) = [SA_1(Z^{l-1}); SA_2(Z^{l-1}); \dots; SA_m(Z^{l-1})]W_O$, where $W_O \in \mathbb{R}^{md \times C}$. The output of each encoder is: $Z^l = MSA(Z^{l-1}) + MLP(MSA(Z^{l-1})) \in \mathbb{R}^{L \times C}$ and total output of all encoders is $\{Z^1, Z^2, \dots, Z^{L_e}\}$. Herein, converting two-dimensional image to one-dimensional sequence may be considered as transforming the image $x \in \mathbb{R}^{C \times H \times W}$ into the sequence $Z \in \mathbb{R}^{L \times C}$. Optionally, when performing feature extraction inference based on the feature extraction model, the extraction process of the second feature vector A (attention maps) may be implemented through general MPC computing engine.

[00127] Optionally, after obtaining the second feature vector A based on the feature extraction model, the second feature vector A may be adjusted (referred to as recalibration based on attention maps in drawings). Herein, when calibrating the second feature vector A , global average pooling (global pooling) may be performed on the second feature vector to obtain the global feature vector information corresponding to each dimension feature of the second feature vector; and based on the softmax probability distribution function (referred to as the SoftMax normalization exponential function (fex(.)) in drawings), the importance weight coefficient s (which may also be denoted by uppercase S) corresponding to each dimension feature of the global feature vector information is obtained. Furthermore, the sigmoid function may be configured to transform the second feature vector A , and the transformed second feature vector (sigmoid(A_k)) is multiplied by corresponding importance weight coefficient s (calibration

product) to obtain the calibrated second feature vector $\tilde{A}$. For example, in the calibrated second feature vector, $\tilde{A}_k = s_k \cdot \text{sigmoid}(A_k)$.

[00128] After obtaining the calibrated second feature vector $\tilde{A}$, the first feature vector F may be calibrated based on the calibrated second feature vector $\tilde{A}$, thereby obtaining the calibrated first baseline feature vector f . For example, XOR operation ($\odot$) is performed on each dimension feature of the second feature vector (attention maps) $\tilde{A}$ and each dimension feature of the first feature vector, and the k -th vector (attention map) after XOR operation: $v_k = \sum_{i,j} F_{i,j,k} \odot \tilde{A}_{i,j,k}$, which may be not described in detail herein. Herein, the process of calibrating the first feature vector through above-mentioned XOR operation may also be called an attention-based pooling process.

[00129] Optionally, after obtaining the first baseline feature vector, the feature extraction model may be trained based on the first baseline feature vector, configured first loss function, and the like. The trained feature extraction model may be deployed online for extracting feature vectors from images such as face images and the like.

[00130] Exemplarily, when using the trained feature extraction model for feature extraction, upon receiving the image to-be-processed for feature extraction, the image to-be-processed may be inputted into the trained feature extraction model; and based on the output of the feature extraction model, the feature vector of the image to-be-processed is obtained.

[00131] In a possible implementation manner, after obtaining the feature vector of the image to-be-processed, a secure multi-party computation process may be performed based on such feature vector. Exemplarily, the feature extraction model may be encrypted and deployed on a server, one or more clients may upload encrypted data such as images and the like, and

multiple parties collaboratively perform inference or extracting feature vectors of images to implement multi-party secure computation process. The process of performing secure multi-party computation based on feature vectors may employ existing technology, which may be not described in detail herein.

[00132] Embodiment two

[00133] To ensure the execution efficiency of secure multi-party computation and reduce the communication latency between different parties, based on above-mentioned embodiments, in embodiments of the present application, the activation function used by the feature extraction model includes a 2quad activation function. Furthermore, the feature extraction model may also be a model obtained based on knowledge distillation.

[00134] In a possible implementation manner, selecting different activation functions may have different impacts on the computational efficiency of the MPC computation engine.

Referring to FIG. 5, FIG. 5 illustrates a diagram of communication durations of different activation functions provided in embodiments of the present application. It may be seen that when the activation function is a Gaussian error linear units (GELU or gelu) activation function, total runtime duration (referred to as “total time” in drawings) is 11 s, where the communication duration between participants (referred to as “communication time” in drawings) is 9.6 s. When the activation function is quad, total runtime duration is 0.5 s, where the communication duration between participants is 9.6 s. When the activation function is softmax, total runtime duration is 40 s, where the communication duration between participants is 34.1 s. When the activation function is 2relu, total runtime duration is 14.7 s, where the communication duration between participants is 12.8 s. When the activation function is 2quad ($\text{Softmax}(x) \approx \frac{(x+c)^2}{\sum (x+c)^2}$) total runtime

duration is 3.2 s, where the communication duration between participants is 1.7 s. It may be seen that when the activation function of the feature extraction model includes the 2quad activation function, the communication duration between participants and total runtime duration may be significantly reduced. Optionally, in addition to including the 2quad activation function, the activation function of the feature extraction model may also include a quad activation function ($\text{GeLU}(x) \approx 0.125x^2 + 0.25x + 0.5$).

[00135] In a possible implementation manner, to improve the inference speed and accuracy of the feature extraction model, above-mentioned feature extraction model may be a small model obtained based on knowledge distillation. For example, a larger feature extraction model (MPCFormer model) may be distilled into a smaller model, and feature extraction may be performed using the smaller model obtained based on knowledge distillation.

[00136] The process of obtaining the feature extraction model based on the framework structure of the MPCFormer model and the training logic of the SETR model is described hereinafter.

[00137] The Transformer model, the foundation of the ChatGPT large-scale model, is a deep neural network model based on self-attention mechanism, which may efficiently process sequential data in parallel. The Transformer model is a neural network model and learns context and thus meaning by tracking relationships in sequential data (such as words in a sentence), which achieves great success in language understanding, especially in large-scale models. Before the Transformer, users need to train neural networks using large labeled-datasets, which are costly and time-consuming to produce. As foundational model, the Transformer model may replace convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in most cases. The Transformer model may be a large encoder and decoder module that processes data

and implemented based on an encoder-decoder framework. Referring to FIG. 6, FIG. 6 illustrates a schematic of a Transformer model provided in embodiments of the present application. The Transformer model is roughly divided into two parts: an encoder and a decoder, corresponding to the left part and the right part in FIG. 6 (as shown in FIG. 6, the left part corresponds to the encoder, and the right part corresponds to the decoder).

[00138] Specifically, the input of the Transformer model includes two parts: input embedding (word embedding conversion) and position-based word embedding (also known as positional encoding). Whether it's text or an image, the numerical representation of the input embedding may be transformed into a vector representation, and the relationship between text or images in high-dimensional space may be obtained. Positional encoding is added after the input embedding, information with different meanings generated by different input positions may be added into the vector, therefore supplementing the positional information. Herein, both the encoder and decoder parts contain inputs, and the structure of the inputs is same for both parts; the only difference is usage during inference. The encoder infers only once, while the decoder, similar to an RNN, iterates repeatedly, continuously generating prediction results.

[00139] The encoder part of the Transformer model is formed by a quantity N of stacked encoder layers, and each encoder layer includes two sub-layers. The first sub-layer includes a multi-head self-attention (MSA) layer, a normalization layer, and a residual connection; the second sub-layer includes a feedforward fully connected layer, a normalization layer, and a residual connection; and so on. Both sub-layers include an adding and normalization (add & norm) layer. Each encoder layer performs one feature extraction operation on the input, that is, the encoding process.

[00140] The decoder of the Transformer model is also formed by a quantity N of stacked encoder layers, but the structure differs slightly from each layer in the encoder. For each layer of the decoder, in addition to two sub-layers in the encoder, the decoder also includes a sub-layer: a masked multi-head attention layer. As shown in FIG. 6, each sub-layer also uses an adding and normalization (add & norm) layer.

[00141] The decoder of the Transformer model performs feature extraction operation on the target based on the encoder's result and previous prediction result, that is, the decoding process. At output, a linear layer (fully connected output layer) is configured to linearly transform the output to specified dimension and perform dimensionality transformation. Furthermore, a Softmax transfer function layer is configured to scale the numbers in the one-dimensional vector to the 0-1 probability range and ensure that the sum of probability value equals 1, thereby obtaining prediction results of the model (output probability).

[00142] Optionally, the Transformer model may be pre-trained on large dataset to gain general understanding and then fine-tuned on small downstream dataset to learn task-specific features.

[00143] The inference process of the Transformer model may be described as a two-party computation (2PC). In 2PC, for example, a user inputs data, and a model provider inputs the Transformer model; and the user and the model provider jointly compute an inference result. Throughout the inference process, 2PC ensures that each party only knows information about own input and result. Transformer inference with participation of multiple MPC parties may be implemented based on secure multi-party computation and knowledge distillation technology. The MPCFormer model employs a trained (or fine-tuned) Transformer model. The MPCFormer model is a Transformer model having low inference latency and high machine learning (ML)

performance and may perform partial approximate computation based on given MPC; and during computation, a high-performance approximate Transformer model may be constructed using knowledge distillation technique. During inference, the MPCFormer model utilizes an MPC engine to implement private model inference.

[00144] Herein, the inference process based on the MPCFormer model may be described as: a convolution computation stage and an inference stage. The convolution computation stage may be described as: $S = \text{MPCFormer}(T, D, A)$, where T is a trained Transformer model, D is an input dataset, and A is an MPC-based computation process. The inference stage may be described as: $y = \text{MPCs}(X)$. In the MPC-based Transformer model inference, the GeLU and Softmax functions are main sources of communication bottlenecks (high communication complexity). Computation of the GeLU function is slow (i.e., time-consuming) because calculation of the error function requires high-order Taylor expansion involving a large number of multiplication operations. Computation of the Softmax function is slow because an exponential function is evaluated through multiple square iterations, which also involves a large number of multiplication operations.

[00145] The SETR model is described hereinafter. SETR is mainly formed by three parts: input $\rightarrow$ transformation $\rightarrow$ output. Referring to FIG. 7, FIG. 7 illustrates a schematic of a SETR model provided in embodiments of the present application, where in FIG. 7, the part (a) mainly illustrates an input preprocessing and feature extraction process of the SETR model, the part (b) mainly illustrates a progressive up-sampling process, and the part (c) mainly illustrates a multi-level feature aggregation process.

[00146] Firstly, original input image needs to be processed into a format supported by the Transformer (SETR). Patch processing is performed on the input image, each two-dimensional

(2D) image patch is converted into a one-dimensional (1D) sequence, and the one-dimensional sequences are collectively input into the model. To encode the spatial information of each patch, a specific embedding may be learned for each local location and added to a linear projection function to form final input sequence. In such way, although the Transformer (SETR) is unordered, corresponding spatial location information (p) may still be retained.

[00147] The one-dimensional sequence is inputted into the SETR model; and the fully connected linear projection layers (linear projection layers) in the SETR model process the one-dimensional sequence to obtain the patch vector (patch embedding) and position information vector (position embedding) corresponding to the patch. The SETR model includes a plurality of transformer layers. Optionally, each transformer layer mainly includes two parts: a multi-head self-attention (MSA) layer and a multilayer perceptron (MLP; and such two parts are connected to a layer normalization (layer norm) process. Herein, layer norm is a normalization technique used in deep neural networks, which normalizes the output of each neuron in the network, thereby ensuring that the output of each layer has similar distribution.

[00148] Feature extraction is achieved by inputting the one-dimensional sequence into the Transformer architecture (SETR model); and the encoder is configured to compress the spatial resolution of original input image and progressively extract higher-level abstract semantic features. The decoder is configured to up-sample the high-level features extracted by the encoder back to original input resolution for pixel-level prediction, thereby achieving image segmentation.

[00149] Referring to the part (b) of FIG. 7, the objective of the decoder is to generate segmentation result on original two-dimensional image ($H \times W$), which needs to reshape the output Z of the encoder from 2D ($H/256 \times W/256 \times C$) to a 3D feature map $H/16 \times W/16 \times C$.

[00150] The decoder may employ progressive up-sampling (PUP). Specifically, considering that direct one-step up-sampling introduces significant noise, a method of alternately performing convolution and up-sampling may be used; and the up-sampling factor is configured to 2 to reduce noise. Original image size is obtained from the Z^{Le} of $H/16 \times W/16$, which needs a total of four up-sampling operations.

[00151] Refer to the part (c) of FIG. 7, the process of multi-layer feature fusion (MLA) is a multi-level feature fusion similar to a feature pyramid, except that the features Z^L are obtained from different Transformer layers and have same resolution. A feature layer $\{Z^m\} (m \in \{Le/M, 2Le/M, \dots, MLe/M\})$ is extracted every Le/M layers, and the features of M paths are extracted. Herein, for each layer, Z^{Le} is reshaped from $HW/256 \times C$ to $H/16 \times W/16 \times C$ and then processed using a three-layer neural network (kernel sizes $1 \times 1, 3 \times 3, 3 \times 3$), the quantity of channels is halved in the first and third layers, and a 4x bilinear interpolation up-sampling is performed after the third layer. Furthermore, starting from the second path, features from previous paths are sequentially fused to enhance information interaction between paths; after another 3×3 convolutional layer, the feature maps from each path are concatenated along the channel dimension and up-sampled using 4x bilinear interpolation to restore original image size.

[00152] Herein, the process of obtaining the feature extraction model based on the MPCFormer model framework and the SETR training logic may be understood as replacing the SETR model framework in the part (a) of FIG. 7 with the MPCFormer model framework, while still using the training logic (operation logic) in the part (b) and the part (c) of FIG. 7 to train the MPCFormer model, thereby obtaining the feature extraction model.

[00153] Embodiment three

[00154] Based on same technical concept, the present application further provides a convolutional neural network training method. Referring to FIG. 8, FIG. 8 illustrates a schematic of a convolutional neural network training process provided in embodiments of the present application. Such process includes following steps.

[00155] At S801, for any obtained second sample image, the second sample image is inputted into the convolutional neural network to-be-trained, and based on the convolutional neural network, the third feature vector of the second sample image is obtained; based on a preset image processing manner, the second sample image is processed into a one-dimensional sequence, the one-dimensional sequence is inputted into trained feature extraction model, and based on the feature extraction model, the fourth feature vector of the second sample image is obtained; and based on the fourth feature vector, the third feature vector is calibrated to obtain calibrated second baseline feature vector, where the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of the semantic vision transformer model SETR.

[00156] In a possible implementation manner, when training the convolutional neural network (CNN), such as a deep residual network (ResNet) and the like, to improve the accuracy of the feature vectors obtained by the convolutional neural network, any second sample image may be obtained from the second sample image set, and the second sample image obtained may be inputted into the convolutional neural network to-be-trained. Herein, the second sample image may be same as or different from the first sample image, which may be not specifically limited in the present application. Exemplarily, the second sample image and the first sample image may be images containing occluded faces, tilted faces or the like.

[00157] Optionally, the convolutional neural network performs feature extraction on the second sample image to obtain the third feature vector F of the second sample image. Herein, for ease of description, both the third feature vector and the first feature vector obtained by the convolutional neural network are denoted by F .

[00158] In a possible implementation manner, the second sample image may also be processed into a one-dimensional sequence based on a preset image processing manner; such one-dimensional sequence is inputted into trained feature extraction model; and based on the feature extraction model, feature extraction is performed on the second sample image to obtain the fourth feature vector of the second sample image. Herein, the feature extraction model in embodiments of the present application is same as the feature extraction models in above-mentioned embodiments, all being models based on the framework structure of the MPCFormer model and the training logic of the SETR model, which may be not described in detail herein. For ease of description, the fourth feature vector and the second feature vector obtained by the feature extraction model are both denoted by A .

[00159] Optionally, to improve the accuracy of the feature vectors obtained by the convolutional neural network, the third feature vector F may be calibrated based on the fourth feature vector A to obtain calibrated second baseline feature vector f . Herein, the process of calibrating the third feature vector F based on the fourth feature vector A is similar to the process of calibrating the first feature vector F based on the second feature vector A in above-mentioned embodiments. For example, XOR operation is performed on each dimension feature of the fourth feature vector and each dimension feature of the third feature vector to obtain the calibrated second baseline feature vector. XOR operation is also performed on each dimension feature of the fourth feature vector and each dimension feature of the third feature vector; obtained vectors

may be normalized; and based on the normalized vectors, the calibrated second baseline feature vector is determined. Moreover, global average pooling may also be performed on the fourth feature vector to obtain global feature vector information corresponding to each dimension feature of the fourth feature vector; based on the probability distribution function, the importance weight coefficient corresponding to each dimension feature of such global feature vector information is obtained; based on the importance weight coefficient corresponding to each dimension feature, the fourth feature vector is calibrated to obtain the calibrated fourth feature vector ($\tilde{A}$); and based on the calibrated fourth feature vector, the third feature vector is calibrated, which may be not described in detail herein.

[00160] At S802, the convolutional neural network is trained based on the second baseline feature vector, the label feature vector carried by the second sample image and configured second loss function.

[00161] In a possible implementation manner, the second sample image may carry a pre-configured label feature vector, where the configuration process of the label feature vector is not specifically limited in the present application. The convolutional neural network may be trained based on the second baseline feature vector f , the label feature vector carried by the second sample image, and the configured second loss function, thereby obtaining a convolutional neural network capable of being deployed online for extracting image feature vectors (such as feature vectors of face images).

[00162] Optionally, the second loss function may include a cross-entropy loss function, a weighted differential regularization loss function, and an L2 regularization loss function. The second loss function may be configured using existing technology, which may be not described in detail herein.

[00163] Optionally, when training the feature extraction model, whether a recognition result is accurate may be determined by whether the second baseline feature vector is consistent with the label feature vector carried by the second sample image. In a specific implementation, if inconsistent, it indicates that the recognition result of the feature extraction model is not accurate, and the parameters of the feature extraction model need to be adjusted, such that the second baseline feature vector obtained based on the feature extraction model gradually approaches the label feature vector, thereby training the feature extraction model.

[00164] In a specific implementation, when adjusting the parameters of the convolutional neural network, a gradient descent algorithm may be configured to backpropagate the gradients of the parameters of the convolutional neural network, thereby training the convolutional neural network.

[00165] In a possible implementation manner, above-mentioned operation may be performed on each second sample image in the second sample set; and when a preset convergence condition is satisfied, the training of the convolutional neural network is considered complete. Herein, the preset convergence condition may be that the quantity of sample images correctly recognized from the second sample images in the second sample set through original convolutional neural network is greater than a set quantity, or the quantity of training iterations reaches a set maximum quantity of iterations, which may be flexibly configured in specific implementation and may be not specifically limited herein.

[00166] In a possible implementation manner, when training original convolutional neural network, the sample images in the sample set may be divided into training sample images and test sample images; and original CNN is first trained based on the training sample images, and the reliability of the trained CNN is verified based on the test sample images.

[00167] In embodiments of the present application, the second sample image may be inputted into the feature extraction model, such that the feature extraction model may extract features from the second sample image; furthermore, the feature extraction model in embodiments of the present application is a model trained using the framework structure of the MPCFormer model and the training logic of the SETR model, such that the feature extraction model may capture more accurate facial feature vectors (the fourth feature vector) based on the self-attention mechanism, and calibrate the third feature vector obtained by the convolutional neural network based on the fourth feature vector to obtain the second baseline feature vector that more accurately represents image features; and the convolutional neural network is trained based on the second baseline feature vector and configured second loss function, thereby improving the accuracy of image feature vectors extracted by the trained convolutional neural network and implementing fast and accurate extraction of high-precision image feature vectors.

[00168] Exemplarily, when using the trained convolutional neural network for feature extraction, upon receiving the image to-be-processed for feature extraction, the image may be inputted into the trained convolutional neural network; and based on the output of the convolutional neural network, the feature vector of the image to-be-processed is obtained. After obtaining the feature vector, a secure multi-party computation process may be performed based on such feature vector. Referring to FIG. 9, FIG. 9 illustrates a schematic of a secure multi-party computation process provided in embodiments of the present application. The convolutional neural network may be deployed on the server; after obtaining the feature vector of the image to-be-processed through the trained convolutional neural network, the feature vector may be securely split into secret slices and sent the secret slices to multiple clients collaborating on secure multi-party computation. For example, the feature vector may be divided into a secret

share 1 and a secret share 2; the secret share 1 may be distributed to the client of the party 1 participating the secure multi-party computation; and the secret share 2 may be distributed to the client of the party 2 participating the secure multi-party computation, and so on. Each client collaboratively extracts features based on the convolutional neural network in a secret sharing manner to perform secure multi-party computation. Such secure multi-party computation manner may prevent the leakage of users' biometric privacy information such as facial features and the like, thereby improving the security of user data. Herein, the process of performing secure multi-party computation based on feature vectors may adopt existing technology, which may be not described in detail herein.

[00169] Embodiment four

[00170] Based on same technical concept, the present application further provides a model training method. Referring to FIG. 10, FIG. 10 illustrates a schematic of a model training process provided in embodiments of the present application. Such process includes following steps.

[00171] At S1001, for any obtained third sample image, the fifth feature vector of the third sample image is obtained based on a convolutional neural network to-be-trained; based on a preset image processing manner, the third sample image is processed into a one-dimensional sequence, the one-dimensional sequence is inputted into the feature extraction model to-be-trained, and based on the feature extraction model, the sixth feature vector of the third sample image is obtained; and based on the sixth feature vector, the fifth feature vector is calibrated to obtain calibrated third baseline feature vector, where the feature extraction model is a model trained based on the framework structure of the MPCFormer model and the training logic of the semantic vision transformer model SETR.

[00172] In a possible implementation manner, the convolutional neural network and the feature extraction model may be trained simultaneously and separately. For example, when training the convolutional neural network and the feature extraction model, to improve the accuracy of the feature vectors obtained by the trained convolutional neural network and the feature extraction model, any third sample image may be obtained from the third sample image set, and the third sample image obtained is inputted into the convolutional neural network to-be-trained. Herein, the third sample image may be same as or different from the first sample image and the second sample image, which may be not specifically limited in the present application. Exemplarily, the third sample image may be an image containing an occluded face or a tilted face or the like.

[00173] Optionally, the convolutional neural network performs feature extraction on the second sample image to obtain the fifth feature vector F of the third sample image. For ease of description, the fifth feature vector, the third feature vector, and the first feature vector obtained by the convolutional neural network are all denoted by F .

[00174] In a possible implementation manner, the third sample image may also be processed into a one-dimensional sequence based on a preset image processing manner; such one-dimensional sequence is inputted into the feature extraction model to-be-trained; and based on the feature extraction model, feature extraction is performed on the third sample image to obtain the sixth feature vector. Herein, the feature extraction model in embodiments of the present application is same as the feature extraction models in above-mentioned embodiments, all being models based on the framework structure of the MPCFormer model and the training logic of the SETR model, which may be not described in detail herein. For ease of description,

the sixth feature vector, the fourth feature vector and the second feature vector obtained by the feature extraction model are all denoted by A.

[00175] Optionally, to improve the accuracy of the feature vectors obtained by the convolutional neural network and the feature extraction model, the fifth feature vector F may be calibrated based on the sixth feature vector A to obtain calibrated third baseline feature vector f. Herein, the process of calibrating the fifth feature vector F based on the sixth feature vector A is similar to the process of calibrating the first feature vector F based on the second feature vector A and the process of calibrating the third feature vector F based on the fourth feature vector A in the previous embodiments. For example, XOR operation may be performed on each dimension feature of the sixth feature vector and each dimension feature of the fifth feature vector to obtain the calibrated third baseline feature vector. XOR operation may also be performed on each dimension feature of the sixth feature vector and each dimension feature of the fifth feature vector; obtained vectors are normalized; and based on normalized vectors, the calibrated third baseline feature vector is determined. Furthermore, global average pooling may also be performed on the sixth feature vector to obtain global feature vector information corresponding to each dimension feature in the six feature vector; based on probability distribution function, the importance weight coefficient corresponding to each dimension feature of such global feature vector information is obtained; based on the importance weight coefficient corresponding to each dimension feature, the sixth feature vector is calibrated to obtain the calibrated sixth feature vector ($A^{\sim}$); and the fifth feature vector is calibrated based on the calibrated sixth feature vector, and so on, which may be not described in detail herein.

[00176] At S1002, the feature extraction model is trained based on the third baseline feature vector, the label feature vector carried by the third sample image, and the configured first

loss function; and the convolutional neural network is trained based on the third baseline feature vector, the label feature vector carried by the third sample image, and the configured second loss function.

[00177] Herein, the process of training the feature extraction model based on the third baseline feature vector, the label feature vector carried by the third sample image, and the configured first loss function is similar to the process of training the feature extraction model based on the first baseline feature vector, the label feature vector carried by the first sample image, and the configured first loss function in above-mentioned embodiments, which may be not described in detail herein.

[00178] The process of training the convolutional neural network based on the third baseline feature vector, the label feature vector carried by the third sample image, and the configured second loss function is similar to the process of training the convolutional neural network based on the second baseline feature vector, the label feature vector carried by the second sample image, and the configured second loss function in above-mentioned embodiments, which may be not described in detail herein.

[00179] Embodiment five

[00180] Based on same technical concept, the present application further provides a feature extraction method. Referring to FIG. 11, FIG. 11 illustrates a schematic a feature extraction process provided in embodiments of the present application. Such process includes following steps.

[00181] At S1101, the image to-be-processed is received.

[00182] At S1102, the feature vector of the image to-be-processed is obtained based on the feature extraction model or the convolutional neural network trained by any above-mentioned method.

[00183] In a possible implementation manner, the method further includes:

[00184] performing secure multi-party computation based on the feature vector.

[00185] Embodiment six

[00186] Based on same technical concept, the present application provides a feature extraction model training apparatus. Referring to FIG. 12, FIG. 12 illustrates a schematic of a feature extraction model training apparatus provided in embodiments of the present application. Such apparatus includes:

[00187] a first calibrating module 1201, configured to, for any obtained first sample image, obtain the first feature vector of the first sample based on a trained convolutional neural network; process the first sample image into a one-dimensional sequence based on a preset image processing manner, input the one-dimensional sequence into the feature extraction model to-be-trained, and obtain the second feature vector of the first sample image based on the feature extraction model; and based on the second feature vector, calibrate the first feature vector to obtain calibrated first baseline feature vector, where the feature extraction model is a model trained based on the framework structure of the MPCFormer model and the training logic of the semantic vision transformer model SETR; and

[00188] a first training module 1202, configured to train the feature extraction model based on the first baseline feature vector, a label feature vector carried by the first sample image, and a configured first loss function.

[00189] In a possible implementation manner, the first calibrating module 1201 is further configured to:

[00190] perform global average pooling on the second feature vector to obtain global feature vector information corresponding to each dimension feature of the second feature vector, and obtain an importance weight coefficient corresponding to each dimension feature of the global feature vector information based on a probability distribution function; based on the importance weight coefficient corresponding to each dimension feature, calibrate the second feature vector to obtain a calibrated second feature vector; and based on the calibrated second feature vector, perform a subsequent step of calibrating the first feature vector based on the second feature vector.

[00191] In a possible implementation manner, the first loss function includes the importance weight coefficient corresponding to each dimension feature.

[00192] In a possible implementation manner, the first calibrating module 1201 is specifically configured to:

[00193] perform an XOR operation on each dimension feature of the second feature vector and each dimension feature of the first feature vector and obtain the calibrated first baseline feature vector.

[00194] In a possible implementation manner, the first calibrating module 1201 is further configured to:

[00195] perform a normalization processing on vectors obtained after the XOR operation; and based on normalized vectors, determine the calibrated first baseline feature vector.

[00196] In a possible implementation manner, activation functions used in the feature extraction model includes a 2quad activation function.

[00197] In a possible implementation manner, the feature extraction model is a model obtained based on knowledge distillation.

[00198] Embodiment seven

[00199] Based on same technical concept, the present application provides a convolutional neural network training apparatus. Referring to FIG. 13, FIG. 13 illustrate a schematic of a convolutional neural network training apparatus provided in embodiments of the present application. Such apparatus includes:

[00200] a second calibrating module 1301, configured to, for any obtained second sample image, input the second sample image into a convolutional neural network to-be-trained, and based on the convolutional neural network, obtain the third feature vector of the second sample image; process the second sample image into a one-dimensional sequence based on a preset image processing manner, input the one-dimensional sequence into a trained feature extraction model, and obtain the fourth feature vector of the second sample image based on the feature extraction model; and based on the fourth feature vector, calibrate the third feature vector to obtain a calibrated second baseline feature vector, where the feature extraction model is a model trained based on the framework structure of the MPCFormer model and the training logic of the semantic vision transformer model SETR; and

[00201] a second training module 1302, configured to train the convolutional neural network based on the second baseline feature vector, the label feature vector carried by the second sample image, and a configured second loss function.

[00202] In a possible implementation manner, the second calibrating module 1301 is specifically configured to:

[00203] perform an XOR operation on each dimension feature of the fourth feature vector and each dimension feature of the third feature vector and obtain the calibrated second baseline feature vector.

[00204] In a possible implementation manner, the second calibrating module 1301 is further configured to:

[00205] perform a normalization processing on vectors obtained after the XOR operation; and based on normalized vectors, determine the calibrated second baseline feature vector.

[00206] Embodiment eight

[00207] Based on same technical concept, the present application provides a model training apparatus. Referring to FIG. 14, FIG. 14 illustrate a schematic of a model training apparatus provided in embodiments of the present application. Such apparatus includes:

[00208] a third calibrating module 1401, configured to, for any obtained third sample image, obtain a fifth feature vector of a third sample image based on a convolutional neural network to-be-trained; process the third sample image into a one-dimensional sequence based on a preset image processing manner, input the one-dimensional sequence into a feature extraction model to-be-trained, and obtain a sixth feature vector of the third sample image based on the feature extraction model; and based on the sixth feature vector, calibrate the fifth feature vector to obtain a calibrated third baseline feature vector, where the feature extraction model is a model trained based on the framework structure of the MPCFormer model and the training logic of the semantic vision transformer model SETR; and

[00209] a third training module 1402, configured to train the feature extraction model based on the third baseline feature vector, the label feature vector carried by the third sample image, and a configured first loss function; and train the convolutional neural network based on the third baseline feature vector, the label feature vector carried by the third sample image, and a configured second loss function.

[00210] Embodiment nine

[00211] Based on same technical concept, the present application provides a feature extraction apparatus. Referring to FIG. 15, FIG. 15 illustrate a schematic of a feature extraction apparatus provided in embodiments of the present application. Such apparatus includes:

[00212] a receiving module 1501, configured to receive the image to-be-processed; and

[00213] an extracting module 1502, configured to obtain the feature vector of the image to-be-processed based on the feature extraction model trained by any method in the first aspect and the third aspect, or the convolutional neural network trained by any method in the second aspect and the third aspect.

[00214] In a possible implementation manner, the apparatus further includes:

[00215] a secure multi-party computation module, configured to perform secure multi-party computation based on the feature vector.

[00216] Embodiment ten

[00217] Based on same technical concept, the present disclosure further provides an electronic device. Referring to FIG. 16, FIG. 16 illustrate a structural schematic of an electronic device provided in embodiments of the present application. The electronic device may include a processor 1601, a communication interface 1602, a memory 1603, and a communication bus

1604, where the processor 1601, the communication interface 1602, and the memory 1603 may communicate with each other through the communication bus 1604.

[00218] The memory 1603 may store a computer program; and when the program is executed by the processor 1601, the processor 1601 may perform steps in any of above-mentioned method embodiments, which may not be described in detail herein.

[00219] The communication bus mentioned in above-mentioned electronic device may be a peripheral component interconnect (PCI) bus or an extended industry standard architecture (EISA) bus or the like. Such communication bus may be divided into address bus, data bus, control bus and the like. For ease of illustration, only one thick line is used to represent the communication bus in FIG. 16, which may not indicate that there is only one bus or one type of bus.

[00220] The communication interface 1602 may be used for communication between above-mentioned electronic device and other devices.

[00221] The memory may include random access memory (RAM) and may also include non-volatile memory (NVM), such as at least one disk storage device. Optionally, the memory may also be at least one storage device located remotely from above-mentioned processor.

[00222] Above-mentioned processor may be a general-purpose processor, including a central processing unit (CPU), a network processor (NP) or the like; and may also be a digital signal processing (DSP), an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), other programmable logic device, a discrete gate or transistor logic device, a discrete hardware component, or the like.

[00223] Embodiment eleven

[00224] Based on same technical concept, embodiments of the present disclosure provide a computer-readable storage medium. The computer-readable storage medium may store a computer program executable by the electronic device. When the program is run on the electronic device, the program may cause the electronic device to perform steps of any above-mentioned method embodiment, which may not be described in detail herein.

[00225] Based on same technical concept, the present disclosure provides a computer program product. The computer program product may include computer program code; and when the computer program code is executed on the computer, the computer program code may cause the computer to implement methods described in any of above-mentioned embodiments applied to the electronic device.

[00226] In above-mentioned embodiments, implementation may be achieved wholly or partially through software, hardware, firmware, or any combination thereof, and may be wholly or partially implemented in the form of a computer program product. The computer program product includes one or more computer instructions; and when the computer instructions are loaded and executed on a computer, the processes or functions described in embodiments of the present disclosure may be wholly or partially generated.

[00227] Those skilled in the art should understand that the embodiments of the present disclosure may be provided as methods, systems, or computer program products. Therefore, the present disclosure may adopt the form of a complete hardware embodiment, a complete software embodiment, or an embodiment combining software and hardware. Moreover, the present disclosure may adopt the form of a computer program product implemented on one or more computer-usable storage media (including but not be limited to disk storage, CD-ROM, optical storage, and the like) containing computer-usable program codes.

[00228] The present disclosure is described with reference to flowcharts and/or block diagrams of methods, devices (systems), and computer program products according to the embodiments of the present disclosure. It may be understood that each process and/or block in the flowcharts and/or block diagrams, and the combination of processes and/or blocks in the flowcharts and/or block diagrams may be implemented by computer program instructions. Such computer program instructions may be provided to the processor of a general-purpose computer, a special-purpose computer, an embedded processor, or other programmable data processing apparatus, such that the instructions executed by the processor of the computer or other programmable data processing apparatus may implement the functions specified in one or more processes in the flowcharts and/or one or more blocks in the block diagrams.

[00229] Such computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to work in a specific manner, such that the instructions stored in the computer-readable memory produce a manufacturing article including the instruction device. The instruction device implements the functions specified in one or more processes in the flowcharts and/or one or more blocks in the block diagrams.

[00230] Such computer program instructions may also be loaded on a computer or other programmable data processing apparatus, such that a series of operation steps are executed on the computer or other programmable apparatus to produce computer-implemented processing. Therefore, the instructions executed by the processor of the computer or other programmable data processing apparatus may implement the functions specified in one or more processes in the flowcharts and/or one or more blocks in the block diagrams.

[00231] Obviously, those skilled in the art may make various changes and modifications to the present disclosure without departing from the spirit and scope of the present disclosure. In such way, if such modifications and variations of the present disclosure fall within the scope of the claims of the present disclosure and their equivalent technologies, the present disclosure may also be intended to include such modifications and variations.

WHAT IS CLAIMED IS:

1. A feature extraction model training method, the method comprising:

for any obtained first sample image, obtaining a first feature vector of a first sample based on a trained convolutional neural network; processing the first sample image into a one-dimensional sequence based on a preset image processing manner, inputting the one-dimensional sequence into a feature extraction model to-be-trained, and obtaining a second feature vector of the first sample image based on the feature extraction model; and based on the second feature vector, calibrating the first feature vector to obtain a calibrated first baseline feature vector, wherein the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and training the feature extraction model based on the first baseline feature vector, a label feature vector carried by the first sample image, and a configured first loss function.

2. The method according to claim 1, after obtaining the second feature vector of the first sample image, and before calibrating the first feature vector based on the second feature vector, the method further including:

performing global average pooling on the second feature vector to obtain global feature vector information corresponding to each dimension feature of the second feature vector, and obtaining an importance weight coefficient corresponding to each dimension feature of the global feature vector information based on a probability distribution function; based on the importance weight coefficient corresponding to each dimension feature, calibrating the second feature vector to obtain a calibrated second feature vector; and based on the calibrated second

feature vector, performing a subsequent step of calibrating the first feature vector based on the second feature vector.

3. The method according to claim 2, wherein the first loss function includes the importance weight coefficient corresponding to each dimension feature.

4. The method according to claim 1, wherein calibrating the first feature vector based on the second feature vector to obtain the calibrated first baseline feature vector includes:

performing an XOR operation on each dimension feature of the second feature vector and each dimension feature of the first feature vector and obtaining the calibrated first baseline feature vector.

5. The method according to claim 4, after performing the XOR operation on each dimension feature of the second feature vector and each dimension feature of the first feature vector and before obtaining the calibrated first baseline feature vector, the method further including:

performing a normalization processing on vectors obtained after the XOR operation; and based on normalized vectors, determining the calibrated first baseline feature vector.

6. The method according to claim 1, wherein activation functions used in the feature extraction model includes a 2quad activation function.

7. The method according to claim 1, wherein the feature extraction model is a model obtained based on knowledge distillation.

8. A convolutional neural network training method, the method comprising:
- for any obtained second sample image, inputting a second sample image into a convolutional neural network to-be-trained, and based on the convolutional neural network, obtaining a third feature vector of the second sample image; processing the second sample image into a one-dimensional sequence based on a preset image processing manner, inputting the one-dimensional sequence into a trained feature extraction model, and obtaining a fourth feature vector of the second sample image based on the feature extraction model; and based on the fourth feature vector, calibrating the third feature vector to obtain a calibrated second baseline feature vector, wherein the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and
- training the convolutional neural network based on the second baseline feature vector, a label feature vector carried by the second sample image, and a configured second loss function.
9. The method according to claim 8, wherein based on the fourth feature vector, calibrating the third feature vector to obtain the calibrated second baseline feature vector includes:
- performing an XOR operation on each dimension feature of the fourth feature vector and each dimension feature of the third feature vector and obtaining the calibrated second baseline feature vector.

10. The method according to claim 9, after performing the XOR operation on each dimension feature of the fourth feature vector and each dimension feature of the first feature vector and before obtaining the calibrated second baseline feature vector, the method further including:
performing a normalization processing on vectors obtained after the XOR operation; and based on normalized vectors, determining the calibrated second baseline feature vector.

11. A model training method, the method comprising:
for any obtained third sample image, obtaining a fifth feature vector of a third sample image based on a convolutional neural network to-be-trained; processing the third sample image into a one-dimensional sequence based on a preset image processing manner, inputting the one-dimensional sequence into a feature extraction model to-be-trained, and obtaining a sixth feature vector of the third sample image based on the feature extraction model; and based on the sixth feature vector, calibrating the fifth feature vector to obtain a calibrated third baseline feature vector, wherein the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and
training the feature extraction model based on the third baseline feature vector, a label feature vector carried by the third sample image, and a configured first loss function; and training the convolutional neural network based on the third baseline feature vector, the label feature vector carried by the third sample image, and a configured second loss function.

12. A feature extraction method, the method comprising:
receiving an image to-be-processed; and

obtaining a feature vector of the image to-be-processed based on the feature extraction model trained by any method according to claims 1-7 and 11 or based on the convolutional neural network trained by any method according to claims 8-11.

13. The method according to claim 12, wherein the method further includes:

performing secure multi-party computation based on the feature vector.

14. A feature extraction model training apparatus, the apparatus comprising:

a first calibrating module, configured to, for any obtained first sample image, obtain a first feature vector of a first sample based on a trained convolutional neural network; process the first sample image into a one-dimensional sequence based on a preset image processing manner, input the one-dimensional sequence into a feature extraction model to-be-trained, and obtain a second feature vector of the first sample image based on the feature extraction model; and based on the second feature vector, calibrate the first feature vector to obtain a calibrated first baseline feature vector, wherein the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and

a first training module, configured to train the feature extraction model based on the first baseline feature vector, a label feature vector carried by the first sample image, and a configured first loss function.

15. A convolutional neural network training apparatus, the apparatus comprising:

a second calibrating module, configured to, for any obtained second sample image, input a second sample image into a convolutional neural network to-be-trained, and obtain a third feature vector of the second sample image based on the convolutional neural network; process the second sample image into a one-dimensional sequence based on a preset image processing manner, input the one-dimensional sequence into a trained feature extraction model, and obtain a fourth feature vector of the second sample image based on the feature extraction model; and based on the fourth feature vector, calibrate the third feature vector to obtain a calibrated second baseline feature vector, wherein the feature extraction model is a model trained based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and

a second training module, configured to train the convolutional neural network based on the second baseline feature vector, a label feature vector carried by the second sample image, and a configured second loss function.

16. A model training apparatus, the apparatus comprising:

a third calibrating module, configured to, for any obtained third sample image, obtain a fifth feature vector of a third sample image based on a convolutional neural network to-be-trained; process the third sample image into a one-dimensional sequence based on a preset image processing manner, input the one-dimensional sequence into a feature extraction model to-be-trained, and obtain a sixth feature vector of the third sample image based on the feature extraction model; and based on the sixth feature vector, calibrate the fifth feature vector to obtain a calibrated third baseline feature vector, wherein the feature extraction model is a model trained

based on a framework structure of a MPCFormer model and a training logic of a semantic vision transformer model SETR; and

a third training module, configured to train the feature extraction model based on the third baseline feature vector, a label feature vector carried by the third sample image, and a configured first loss function; and train the convolutional neural network based on the third baseline feature vector, the label feature vector carried by the third sample image, and a configured second loss function.

17. A feature extraction apparatus, the apparatus comprising:

a receiving module, configured to receive an image to-be-processed; and

an obtaining module, configured to obtain a feature vector of the image to-be-processed based on the feature extraction model trained by any method according to claims 1-7 and 11 or based on the convolutional neural network trained by any method according to claims 8-11.

18. An electronic device, wherein the electronic device at least comprises a processor and a memory; and when executing a computer program stored in the memory, the processor is configured to implement steps of any method according to claims 1-13.

19. A computer-readable storage medium, storing a computer program, wherein when executed by a processor, the computer program implements steps of any method according to claims 1-13.