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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 May 2017 (04.05.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/070858 A1

(51) International Patent Classification:
G06K 9/00 (2006.01)

(21) International Application Number:
PCT/CN2015/093031

(22) International Filing Date:
28 October 2015 (28.10.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicants: **BEIJING SENSETIME TECHNOLOGY DEVELOPMENT CO., LTD** [CN/CN]; Room 710-712, 7th Floor, 3rd Building, 1st Courtyard, Zhongguancun East Road, Haidian District, Beijing 100084 (CN). **SHENZHEN SENSETIME TECHNOLOGY CO., LTD** [CN/CN]; Room 201, Building A, No. 1, 1st Qianwan Road, Qianhai Shenzhen-Hongkong Modern Service Industry Cooperation Zone, Shenzhen, Guangdong 518000 (CN). **SENSETIME GROUP LIMITED** [CN/CN]; Units 226 to 230, 2/F., Core Building 2, No. 1 Science Park West Avenue, Hong Kong Science Park, Shatin, N.T., Hong Kong (CN).

(72) Inventors: **SUN, Yi**; Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (CN). **WANG, Xiaogang**; Department of Electrical Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (CN). **TANG, Xiaoou**; Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (CN).

(74) Agent: **INSIGHT INTELLECTUAL PROPERTY LIMITED**; 19 A, Tower A, InDo Building, No. 48A Zhichun Road, Haidian District, Beijing 100098 (CN).

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

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

[Continued on next page]

(54) Title: A METHOD AND A SYSTEM FOR FACE RECOGNITION

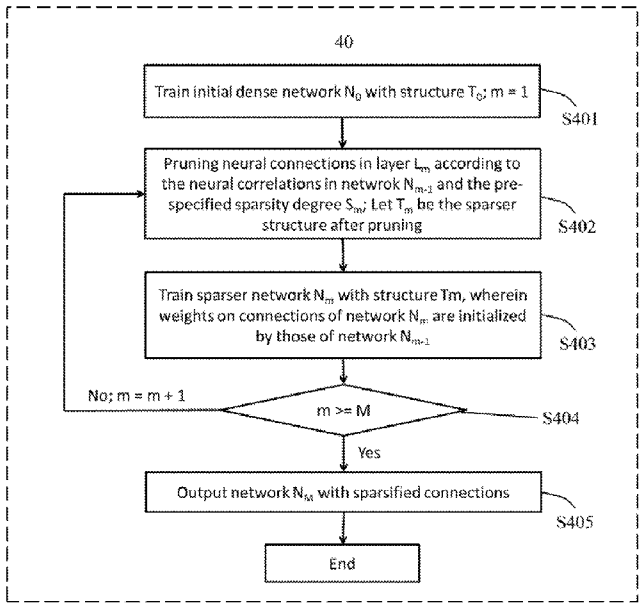

Fig. 4

(57) Abstract: Disclosed is an apparatus for face recognition, comprising: a feature extraction unit configured to extract features from input face images with a plurality of deep feature extraction hierarchies; and a recognition unit configured to calculate distances between facial features of different face images extracted by the extractor to determine if two face images are from the same identity for face verification or determine if one of the input images, as a probe face image, is belonging to a same identity as one of gallery face images consisting of the input images for face identification, wherein each of the deep feature extraction hierarchies contains a plurality of cascaded convolution layers, local-connection layers, pooling layers, and full-connection layers, and wherein neurons in the full-connection layers are only connected to a part of neurons in a previous layer thereof, while neurons in the convolution layers and the local-connection layers are only connected to a part of neurons in local regions in a previous layer thereof.

WO 2017/070858 A1

Published:

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

A METHOD AND A SYSTEM FOR FACE RECOGNITION

Technical Field

[0001] The present application relates to a method for face recognition and a system thereof.

Background

[0002] The number of parameters in a deep neural network is restricted by the amount of training data. Weight sparsifying algorithms help to reduce model parameters and improve the generalization ability of deep models.

[0003] The idea of reducing neural connections has been taken in designing GoogLeNet, which achieved great success on the ImageNet challenge. GoogLeNet reduced neural connections by using very small convolution kernels of sizes 1x1, 3x3, and 5x5.

[0004] The *Hebbian* rule of "neurons that fire together wire together" suggests that connections between strongly correlated neurons are more important than those between weakly correlated neurons. Moreover, neurons in the previous layer which are more correlated (either positively or negatively) to a given neuron in the current layer are more helpful to predict the activities of the latter.

[0005] Removing unimportant parameters in deep neural networks was studied by *LeCun et al.* in their seminal work *Optimal Brain Damage*. They took a second derivative-related criterion for removing parameters. They reduced model parameters by a factor of eight without loss of the prediction ability of the original model.

Summary

[0006] In one aspect of the present application, disclosed is an apparatus for face recognition. The apparatus may comprise an extractor having a plurality of deep

neural networks with sparsified neural connections to extract facial features from multiple face regions of face images for face recognition; and a recognizer being electronically communicated with the extractor and recognizing face identities of the input face images based on the extracted facial features.

[0007] Guided by the Hebbian rule that "neurons that fire together wire together", more neural connections between weekly correlated neurons than those between strongly correlated neurons are pruned, wherein the correlation between two connected neurons are defined by the magnitude of the correlation between their neural activations.

[0008] In one embodiment of the present application, a baseline deep neural network is first trained, and then neural connections are pruned layer-wisely from the last to the previous layers, each time only one additional layer is sparsified and the entire model is re-trained. The previously trained models are used to calculate the neural correlations and initialize the subsequent sparser models.

[0009] In one embodiment of the present application, the baseline deep neural network is similar to VGG net with every two convolutional layers following one max-pooling layer. One major difference is that the last two convolutional layers are replaced by two locally-connected layers. The aim is to learn different features in different face regions, since face is a structured object, and local connections increase the model fitting ability. The second locally-connected layer is followed by a multi-dimensional fully-connected layer. The feature representation in the fully-connected layer is used for the following face recognition.

[0010] In one embodiment of the present application, connections in the baseline model are deleted in a layer-wise fashion, from the last fully-connected layer to the previous locally-connected and convolutional layers. Let N_0 denote a well-trained baseline model. When a layer L_m is sparsified, a new model N_m is re-trained

initialized by its previous model N_{m-1}. Therefore, a sequence of models $\{N_1, \ldots, N_M\}$ with fewer and fewer connections are trained and N_M is the final sparse ConvNet obtained. During the whole training process, the previously learned model is used to calculate the neural correlations and guide the connection dropping procedure. The weights learned by the denser model N_{M-1} are also good initialization of the sparser model N_m to be further trained.

[0011] In some of embodiments, a trainer may be electronically communicated with the extractor to add supervisory signals on the deep neural networks during training so as to learn sparse structures in convolution, local-connection, and full-connection layers, as well as adjusting neural weights in these layers.

[0012] In one embodiment of the present application, joint identification-verification supervisory signal is added to the last fully-connected layer. The same supervisory signal is also added to a few previous layers to enhance the supervision in previous feature learning stages. The supervisory signals comprise one identification supervisory signal and one verification supervisory signal, wherein the identification supervisory signal is generated by classifying features in any of the layers extracted from an input face region into one of N identities in a training dataset, and taking a classification error as the supervisory signal, and wherein the verification signal is generated by comparing features in any of the layers extracted from two input face images respectively for determining if they are from the same person, and taking a verification error as the supervisory signal.

[0013] Neural weights and neural connections are updated alternatively and iteratively. Firstly, neural connections are fixed while neural weights are adjusted by back-propagating supervisory signals through the deep neural networks. These supervisory signals are aggregated to adjust neural weights in each of convolution, local-connection, and full-connection layers during training. Then neural weights are fixed while neural connections are pruned according to correlations between neural

activations of connected neurons. The majority of weakly correlated neurons are pruned. Given the sparser deep model, neural weights are updated again by fixing neural connections, and so forth.

[0014] In further aspect of the present application, disclosed is a method for face recognition, comprising: configuring a plurality of deep feature extraction hierarchies such that each of the deep feature extraction hierarchies contains a plurality of cascaded convolution layers, local-connection layers, pooling layers and full-connection layers, and neurons in the full-connection layers are only connected to a part of neurons in a previous layer thereof, while neurons in the convolution layers and the local-connection layers are only connected to a part of neurons in local regions in a previous layer thereof; training the configured deep feature extraction hierarchies to learn neural connections in the convolution layers, the local-connection layers, and the full-connection layers, and to adjust neural weights in these layers; extracting features from input face images by the trained deep feature extraction hierarchies; and recognizing face identities of the input face images based on the extracted facial features.

Brief Description of the Drawing

[0015] Exemplary non-limiting embodiments of the present invention are described below with reference to the attached drawings. The drawings are illustrative and generally not to an exact scale. The same or similar elements on different figures are referenced with the same reference numbers.

[0016] Fig. 1 is a schematic diagram illustrating an apparatus for face recognition consistent with some disclosed embodiments.

[0017] Fig.2 is a schematic diagram illustrating an apparatus for face recognition when it is implemented in software, consistent with some disclosed embodiments.

[0018] Fig. 3 is a schematic diagram illustrating an example of deep neural networks with sparsified layers in the extractor as shown in Fig. 1.

[0019] Fig. 4 is a schematic flowchart illustrating the trainer as shown in Fig. 1 consistent with some disclosed embodiments.

[0020] Fig. 5 is a schematic flowchart illustrating the extractor as shown in Fig. 1 consistent with some disclosed embodiments.

[0021] Fig. 6 is a schematic flowchart illustrating the recognizer as shown in Fig. 1 consistent with some disclosed embodiments.

Detailed Description

[0022] Reference will now be made in detail to some specific embodiments of the invention including the best modes contemplated by the inventors for carrying out the invention. Examples of these specific embodiments are illustrated in the accompanying drawings. While the invention is described in conjunction with these specific embodiments, it will be understood that it is not intended to limit the invention to the described embodiments. On the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. The present invention may be practiced without some or all of these specific details. In other instances, well-known process operations have not been described in detail in order not to unnecessarily obscure the present invention.

[0023] As will be appreciated by one skilled in the art, the present invention may be embodied as a system, method or computer program product. Accordingly, the present invention may take the form of an entirely hardware embodiment, an entirely

software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, the present invention may take the form of a computer program product embodied in any tangible medium of expression having computer-usable program code embodied in the medium.

[0024] In the case that the apparatus 1000 as disclosed below is implemented with software, the apparatus 1000 may include a general purpose computer, a computer cluster, a mainstream computer, a computing device dedicated for providing online contents, or a computer network comprising a group of computers operating in a centralized or distributed fashion. As shown in Fig. 2, the apparatus 1000 may include one or more processors (processors 102, 104, 106 etc.), a memory 112, a storage device 116, a communication interface 114, and a bus to facilitate information exchange among various components of apparatus 1000. Processors 102-106 may include a central processing unit ("CPU"), a graphic processing unit ("GPU"), or other suitable information processing devices. Depending on the type of hardware being used, processors 102-106 can include one or more printed circuit boards, and/or one or more microprocessor chips. Processors 102-106 can execute sequences of computer program instructions to perform various methods or run the modules that will be explained in greater detail below.

[0025] Memory 112 can include, among other things, a random access memory ("RAM") and a read-only memory ("ROM"). Computer program instructions can be stored, accessed, and read from memory 112 for execution by one or more of processors 102-106. For example, memory 112 may store one or more software applications. Further, memory 112 may store an entire software application or only a part of a software application that is executable by one or more of processors 102-106 to carry out the functions as disclosed below for the apparatus 1000. It is noted that

although only one block is shown in Fig. 1, memory 112 may include multiple physical devices installed on a central computing device or on different computing devices.

[0026] Referring to Fig. 1 again, where the apparatus 1000 is implemented by the hardware, it may comprise an extractor 10 and a recognizer 20. The extractor 10 is configured with a plurality of deep neural networks with sparsified neural connections (referred to as sparse deep neural networks) to extract facial features from face regions of input face images. The recognizer 20 is electronically communicated with the extractor 10 and recognizes face identities of the input face images based on the extracted facial features. As will be discussed in details below, each of the sparse deep neural networks comprises a plurality of sparse convolutional layers, sparse local-connection layers, sparse full-connection layers, and pooling layers. A first one of the sparse convolutional layers extracts local facial features from input face images, and the followings of the sparse convolutional layers and sparse local-connection layers extract further local features from the extracted features outputted from a previous layer. Each of sparse full-connection layers extract global features from the extracted features outputted from a previous layer. Each of pooling layers receives features from a previous layer and reduces dimensions of the received features. The features obtained from all the sparse deep neural networks are concatenated into a feature vector as said facial features for face recognition.

[0027] In addition, the apparatus 1000 may further comprise a trainer 30 used to learn sparse neural connections (referred to as sparse structures) as well as weights on sparse connections of the sparse deep neural networks.

The Extractor 10

[0028] Fig. 5 is a schematic flowchart illustrating the feature extraction process 50 in the extractor 10, which contains three steps. In step S501, the extractor 10 forward propagates face regions of an input face image through deep neural networks with

sparsified connections (referred to as sparse deep neural networks). Then in step S502, the extractor 10 takes neural activations in last layers of sparse deep neural networks as facial features. Finally in step S503, it concatenates facial features of all sparse deep neural networks.

[0029] Given an input face image, the sparse deep neural network in the extractor 10 starts by extracting local facial features with every two sparse convolutional layers following one pooling layer. The last pooling layer (pooling layer 12) is followed by two sparse local-connection layers to further extract local facial features and one sparse full-connection layer to extract global facial features. In particular, in the sparse deep neural network, wherein neurons in the full-connection layers are only connected to a part of neurons in a previous layer thereof, while neurons in the convolution layers and the local-connection layers are only connected to a part of neurons in local regions in a previous layer thereof. This is why these layers are called as "sparse" layers.

[0030] As will be discussed later in reference to the trainer 30, the sparse deep neural network in the extractor 10 shall be trained. In one embodiment of the present application, connections in the baseline model are deleted in a layer-wise fashion, from the last fully-connected layer to the previous locally-connected and convolutional layers. Let N_0 denote a well-trained baseline model. When a layer L_m is sparsified, a new model N_m is re-trained initialized by its previous model N_{m-1}. Therefore, a sequence of models $\{N_1, \ldots, N_M\}$ with fewer and fewer connections are trained and N_M is the final sparse deep neural network (also referred to as sparse ConvNet since the deep neural network contains convolutional layers) obtained. During the whole training process, the previously learned model is used to calculate the neural correlations and guide the connection dropping procedure. The weights learned by the denser model N_{M-1} are also good initialization of the sparser model N_m to be further trained.

[0031] With the above constructions, by reducing model/layer parameters (i.e., neural weights on connections) of the original non-sparse layers, these sparse layers help to improve the generalization ability of the learned features, i.e., features learned on the training face images can be well generalized to test face images to distinguish the test face images well by their identities. In addition, the sparse layers reduce sizes (parameters) of neural networks, making it easier to be stored on mobile phones or other devices with limited memories.

[0032] Fig. 3 illustrates an example of sparse deep neural networks in the extractor 10 according to one embodiment of the present application. The extractor 10 contains a plurality of sparse deep neural networks. Each of the sparse deep neural networks may comprise a plurality of sparse convolution-pooling modules 301, 302, 303 and 304, and comprises a connection module 305, as illustrated in Fig. 3. It will be appreciated that there would be more or less number of sparse convolution-pooling modules as required, although Fig. 3 illustrates 4 sparse convolution-pooling modules as an example.

[0033] As shown, each of the sparse convolution-pooling modules 301, 302, 303 and 304 is a cascade of two sparse convolutional layers and a pooling layer. For example, the convolution-pooling modules 301 may comprise a sparse convolution layer 1, a sparse convolution layer 2 and a pooling layer 3, which are cascaded sequentially. All the sparse convolution-pooling modules 301, 302, 303 and 304 are cascaded sequentially, and then are cascaded to the connection module 305 which further configured with two sparse local-connection layers 13 and 14, and a sparse full-connection layer 15. Compared to convolutional layers, local-connection layers 13 and 14 help to extract more diverse features, which are proved to be helpful in later feature extraction stages in deep neural networks. Neural activations in sparse full-connection layer 15 are used as facial features for face recognition.

[0034] Sparse convolutional layers, sparse local-connection layers, and sparse

full-connection layers are convolutional layers, local-connection layers, and full-connection layers with sparsified neural connections, respectively. Given the degree of sparsity S ($0 < S < 1$), the present application samples $S \cdot |W|$ weights from the total number of weights $|W|$ in a given sparse layer. Neural connections which correspond to the sampled weights are reserved. Otherwise, they are pruned from the current sparse deep neural network. The number of connections is proportional to the number of weights for all types of sparse layers in sparse deep neural networks.

[0035] Convolutional layers are configured to extract local facial features from input feature maps (which is output feature maps of a previous layer) to form output feature maps of the current layer. In particular, each convolutional layer performs convolution operations on the input feature maps to form output feature maps of the current layer, and the formed output feature maps will be input to a next layer.

[0036] Each feature map is a certain kind of features organized in 2D. Features in the same output feature map are extracted from input feature maps with the same set of neural connection weights. The convolution operation in each convolutional layer may be expressed as

$$y^j = \max\left(0, b^j + \sum_i k^{ij} * x^i\right) \qquad (1)$$

Where,

x^i and y^j are the i-th input feature map and the j-th output feature map, respectively;

k^{ij} is the convolution kernel between the i-th input feature map and the j-th output feature map;

$*$ denotes convolution;

b^j is the bias of the j-th output feature map;

ReLU nonlinearity $y = \max(0, \cdot)$ is used for neurons..

[0037] Neural weights in convolutional layers are parameters in convolution kernels k^{ij}. In sparse convolutional layers, a portion of kernel parameters are sampled,

according to the degree of sparsity S. A sampled parameter corresponds to a set of neural connections which share the same parameter. These neural connections are reserved in sparse convolutional layers. Other neural connections which take the unsampled kernel parameters as weights are pruned from a sparse convolutional layer.

[0038] Local-connection layers are also configured to extract local facial features from input feature maps (which is output feature maps of a previous layer) to form output feature maps of the current layer. Unlike convolutional layers, local-connection layers do not share neural weights across neurons on the same output feature map. The operation in each local-connection layer may be expressed as

$$y^{jr} = \max\left(0, b^{j} + \sum_{i} k^{ijr} \cdot x^{ir}\right) \qquad (2)$$

Where, x^{ir} is neural activations in a local region r in the i-th feature map of a previous layer. y^{jr} is the r-th (single) neural activation in the j-th output feature map of the current layer. k^{ijr} is neural weights on local connections between y^{jr} and x^{ir}. b^{j} is the bias of the j-th output feature map. $y = \max(0, \cdot)$ is the ReLU nonlinearity. In sparse local-connection layers, a portion of neural weights (i.e., k^{ijr} for all i, j, and r) are sampled, according to the degree of sparsity S. A sampled neural weight corresponds to a single neural connection since weights are unshared between different neural connections. These sampled neural connections are reserved in sparse local-connection layers. Other unsampled neural connections are pruned from sparse local-connection layers.

[0039] The goal of cascading (sparse) convolutional layers and (sparse) local-connection layers is to extract hierarchical local features (i.e. features extracted from local regions of the input images or the input features), wherein features extracted by higher convolutional/local-connection layers have larger effective receptive field on input images and more complex non-linearity.

[0040] Pooling layers are configured to pool local facial features from input feature maps from a previous layer to form output feature maps of the current layer. Pooling

operation forms more invariant features, which is formulated as

$$y_{j,k}^{i} = \max_{\substack{0 \le m < M \\ 0 \le n < N}} \left\{ x_{j \cdot s+m, k \cdot s+n}^{i} \right\} \qquad (3)$$

where each neuron in the i-th output feature map y^i pools over an $M \times N$ local region in the i-th input feature map x^i, with s as the step size.

[0041] Full-connection layers in deep neural networks are configured to extract global features (features extracted from the entire region of input feature maps) from a previous layer. Full-connection layers also serve as interfaces for receiving supervisory signals during training, which will be discussed later. Full-connection layers also have the function of feature dimension reduction as pooling layers by restricting the number of neurons in them. A full-connection layer is formulated as

$$y_j = \max\left(0, \sum_i x_i \cdot w_{i,j} + b_j\right) \qquad (4)$$

Where,

x denotes neural activations from a previous layer,

y denotes neural activations in the current full-connection layer,

w denotes neural weights on connections between the current full-connection layer and a previous layer. Neurons in full-connection layers linearly combine neural activations of all neurons in a previous layer, followed by ReLU non-linearity. In sparse full-connection layers, a portion of neural weights (i.e., $w_{i,j}$ for all i and j) are sampled, according to the degree of sparsity S. A sampled neural weight corresponds to a single neural connection since weights are unshared between different neural connections. These sampled neural connections are reserved in sparse full-connection layers. Other unsampled neural connections are pruned from sparse full-connection layers.

[0042] Neural activations of neurons in the highest layer of sparse deep neural networks are used as facial features for face recognition. These facial features are global and can capture highly non-linear mappings from input face images to their identities. In one embodiment of the present application, neural activations of neurons

in sparse full-connection layer 15 of the sparse deep neural network shown in Fig. 3 are used as facial features for face recognition. An extractor contains a plurality of sparse deep neural networks. Facial features extracted by all sparse deep neural networks are concatenated into a long feature vector as a final feature representation for face recognition.

The recognizer 20

[0043] The recognizer 20 operates to calculate distances between facial features of different face images extracted by the extractor 10 to determine if two face images are from the same identity for face verification or determine if one of the input images, as a probe face image, is belonging to a same identity as one of gallery face images consisting of the input images for face identification. Fig. 6 is a schematic flowchart illustrating the recognition process 60 in the recognizer 20. In step S601, the recognizer 20 calculates distances between facial features extracted from different face images by the extractor 10. Then in step S602, the recognizer 20 determines if two face images are from the same identity for face verification, or, alternatively, in step S603, it determines one of the input images, as a probe face image, is belonging to a same identity as one of gallery face images consisting of the input images for face identification.

[0044] In the recognizer 20, two face images are determined to belong to the same identity if their feature distance is smaller than a threshold, or the probe face image is determined to belong to the same identity as one of gallery face images if their feature distance is the smallest compared to feature distances of the probe face image to all the other gallery face images, wherein feature distances determined by the recognizer 20 could be Euclidean distances, Joint Bayesian distances, cosine distances, Hamming distances, or any other distances.

The Trainer 30

[0045] The trainer 30 is used to learn the sparse structures (i.e., neural connections) of

the sparse deep neural networks as well as neural weights on connections of the sparse deep neural networks in the extractor 10. As illustrated in Fig. 4, in step S401 the trainer 30 first trains an initial dense neural network N_0 with network structure T_0. For example, the initial structure T_0 could be the one shown in Fig. 3 by replacing the conventional convolutional layers, the conventional local-connection layers, and the conventional full-connection layers with the sparse convolutional layers, the sparse local-connection layers, and the sparse full-connection layers, respectively. Then, given a sequence of layers L_1, L_2, …, L_M to be sparsified and the corresponding pre-specified degrees of sparsify S_1, S_2, …, S_M, the trainer 30 iteratively prunes (sparsifies) the neural connections as illustrated in step S402 and learns the neural weights on reserved neural connections as illustrated in step S403. In the m-th iteration (for m = 1, 2, …, M), the trainer 30 first in step S402 prunes the neural connections in layer L_m according to the neural correlations in netwrok N_{m-1} and the pre-specified sparsity degree S_m. Let T_m be the sparser structure after pruning. The trainer 30 then in step S403 trains a sparser network N_m with structure T_m, wherein weights on connections of network N_m are initialized by those of network N_{m-1}. After iterative connection pruning (sparsifying) and weight updating (m > = M, at step S404), the trainer 30 finally in step S405 outputs the sparsified and well-trained neural network N_M.

[0046] Given a layer L_m to be pruned (sparsified) and the pre-specified degree of sparsity S_m ($0 < S_m < 1$) for the given layer, the present application samples $S_m \cdot |W|$ weights from the total number of weights $|W|$ of layer L_m. The number of connections is proportional to the number of weights for all types of sparse layers (including sparse convolutional layers, sparse local-connection layers, and sparse full-connection layers) in the sparse deep neural network. The sampling is based on neural correlations with such a principle that keeps connections (and the corresponding weights) where neurons connected have high correlations and drops connections between weakly correlated neurons. This is because neurons in one layer which have stronger correlations to neurons in the upper layer have stronger predictive power for

the activities of the latter. Note that neurons with strong negative correlations are also useful in predicting neural activations. If a neuron is viewed as a detector of a certain visual pattern, its positively correlated neurons in the lower layer provide evidence on the visual pattern, while its negatively correlated neurons help to reduce false alarms. In practice, the present application may also keep a small portion of connections between weakly correlated neurons due to the reason that predictions from weakly correlated neurons are complementary to those from highly correlated neurons.

[0047] First, the full-connection layers and local-connection layers in which weights are not shared are considered. Weights and connections are one-to-one mapped in these layers. Given a neuron a_i in the current layer and its K connected neurons b_{i1}, b_{i2}, …, b_{iK} in the previous layer, the correlation coefficient between a_i to each of b_{ik} for k = 1, 2, …, K is (for simplicity, when the present application refers to a neuron, it also means its neural activations)

$$r_{ik} = \frac{E[a_i - \mu_{a_i}][b_{ik} - \mu_{b_{ik}}]}{\sigma_{a_i}\sigma_{b_{ik}}} \quad (5)$$

where μ_{a_i}, $\mu_{b_{ik}}$, σ_{a_i}, and $\sigma_{b_{ik}}$ denote the mean and standard deviation of a_i and b_{ik}, respectively, which are evaluated on a separate training set. Since both positively and negatively correlated neurons are helpful for the predictions, the corresponding connections are considered respectively. From all r_{ik} for k = 1, 2, …, K, first take out all positive coefficients and sort them in descending order, denoted as r_{ik}^{+} for k = 1, 2, …, K^+. Then randomly sample λSK^+ and $(1-\lambda)SK^+$ coefficients from the coefficients ranked in the first and the second half of the sorted correlation coefficients, respectively. Weights/connections corresponding to the sampled coefficients are reserved while others are deleted. The present application takes $\lambda = 0.75$. In other words, connections from the half of higher correlations are three times as much as those from the half of lower correlations. The total kept connections/weights are SK^+, which depends on the degree of sparsity S.

[0048] The negative coefficients are processed in a similar way, except that the

absolute value of the coefficients are considered and more coefficients (and the corresponding connections/weights) of higher absolute values are kept. The total sampled negative coefficients are SK^-, given K^- negative coefficients from r_{ik} for k = 1, 2, …, K. Connections from each of output neurons a_i are processed in the same way. Suppose there are N output neurons a_i for i = 1, 2, …, N. Then the total sampled weights/connections are SKN.

[0049] For convolutional layers, the set of correlation coefficients between neurons with shared connecting weights are jointly considered to determine whether a weight (or a set of connections with shared weights) should be reserved or deleted. Let a_{im} be the m-th neuron in the i-th feature map of the current layer, and it is connected to K neurons b_{mk} in the previous layer for k = 1, 2, …, K. (K equals the filter size, e.g., 3x3, times the number of input channels.) The set of K neurons b_{mk} are determined by the position m. There are a total of M neurons in the i-th output feature map as a_{im} for m = 1, 2, …, M. They all share the same set of K weights, although connected to different sets of neurons in the previous layer b_{mk} for m = 1, 2, …, M. Weights between a_{im} and b_{mk} are shared for m = 1, 2, …, M. We calculate the mean magnitude of the correlation coefficients between a_{im} and b_{mk} for m = 1, 2, …, M as

$$r_{ik} = \sum_{m=1}^{M} \left| \frac{E[a_{im} - \mu_{a_{im}}][b_{mk} - \mu_{b_{mk}}]}{\sigma_{a_{im}} \sigma_{b_{mk}}} \right| \quad (6)$$

Similar to the case in the full-connection layers and local-connection layers, given the degree of sparsity S, SK mean correlation coefficients (and the corresponding weights) from the set of K coefficients r_{ik} for k = 1, 2, …, K are selected. r_{ik} are sorted in descending order. λSK coefficients are randomly chosen from the first half with higher values and $(1-\lambda)SK$ coefficients are randomly chosen from the second half of the correlation coefficients with lower values. Again λ is set to 0.75 in the present application. The set of K weights r_{ik} for k = 1, 2, …, K are processed in the same way for all i = 1, 2, …, N (given N feature maps in the current layer). The total sampled weights are SKN.

[0050] During the phase of weight updating, identification and verification supervisory signals in the trainer 30 are simultaneously added to each of the supervised layers (e.g., sparse full-connection layer 15 in Fig. 3) of each of the sparse deep neural networks in the extractor 10, and respectively back-propagated to the input face image, so as to update neural weights on reserved neural connections of sparse convolutional layers, sparse local-connection layers, and sparse full-connection layers of the sparse deep neural networks.

[0051] The identification supervisory signals are generated in the trainer 30 by classifying all of the supervised layer (layers selected for supervision, e.g., sparse full-connection layer 15 in Fig. 3) representations into one of N identities, wherein the classification errors are used as the identification supervisory signals.

[0052] The verification supervisory signals in the trainer 30 are generated by verifying the supervised layer representations of two compared face images, respectively, in each of the feature extraction modules, to determine if the two compared face images belong to the same identity, wherein the verification errors are used as the verification supervisory signals. Given a pair of training face images, the extractor 10 extracts two feature vectors f_i and f_j from the two face images respectively in each of the feature extraction modules. The verification error is $\frac{1}{2}\left\|f_i - f_j\right\|_2^2$ if f_i and f_j are features of face images of the same identity, or $\frac{1}{2}\max\left(0, m - \left\|f_i - f_j\right\|_2\right)^2$ if f_i and f_j are features of face images of different identities, where $\left\|f_i - f_j\right\|_2$ is Euclidean distance of the two feature vectors, m is a positive constant value. There are errors if f_i and f_j are dissimilar for the same identity, or if f_i and f_j are similar for different identities.

[0053] Although the preferred examples of the present invention have been described,

those skilled in the art can make variations or modifications to these examples upon knowing the basic inventive concept. The appended claims is intended to be considered as comprising the preferred examples and all the variations or modifications fell into the scope of the present invention.

[0054] Obviously, those skilled in the art can make variations or modifications to the present invention without departing the spirit and scope of the present invention. As such, if these variations or modifications belong to the scope of the claims and equivalent technique, they may also fall into the scope of the present invention.

[0055] The corresponding structures, materials, acts, and equivalents of all means or step plus function elements in the claims below are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed. The description of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the invention. The embodiment was chosen and described in order to best explain the principles of the invention and the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. An apparatus for face recognition, comprising:

a feature extraction unit configured to extract features from input face images with a plurality of deep feature extraction hierarchies; and

a recognition unit configured to calculate distances between facial features of different face images extracted by the extractor to determine if two face images are from the same identity for face verification or determine if one of the input images, as a probe face image, is belonging to a same identity as one of gallery face images consisting of the input images for face identification,

wherein each of the deep feature extraction hierarchies contains a plurality of cascaded convolution layers, local-connection layers, pooling layers, and full-connection layers, and

wherein neurons in the full-connection layers are only connected to a part of neurons in a previous layer thereof, while neurons in the convolution layers and the local-connection layers are only connected to a part of neurons in local regions in a previous layer thereof.

2. An apparatus of claim 1, further comprising:

a training unit configured to add supervisory signals on the feature extraction unit during training so as to learn neural connections in the convolution layers, the local-connection layers, and the full-connection layers, and to adjust neural weights in these layers.

3. An apparatus of claim 1, wherein neural connections in the convolution layers, the local-connection layers, and the full-connection layers and neural weights on the neural connections are learned iteratively.

4. An apparatus of claim 3, wherein in one iteration, the neural weights on neural

connections are adjusted by fixing neural connections in the convolution, local-connection and full-connection layers, and then,

the neural connections in one or more of the convolution, local-connection and full-connection layers are pruned while fixing neural weights.

5. An apparatus of claim 3, wherein the neural connections are pruned according to correlations between neural activations of connected neurons, wherein a majority of connections between weakly correlated neurons are pruned while a majority of connections between strongly correlated neurons are reserved.

6. An apparatus of claim 3, wherein before the first iteration, neurons in the full-connection layers are connected to all neurons in a previous layer thereof, while neurons in the sparse convolution modules and the sparse local-connection modules are connected to all neurons in local regions in a previous layer thereof, respectively.

7. An apparatus of claim 3, wherein for the second and the following iterations, the neuron connections are taken from reserved neuron connections in the previous iteration and the neuron weights on these neuron connections are initialized by neuron weights learned in the previous iteration.

8. An apparatus of claim 7, wherein the neuron weights on the reserved neuron connections from the previous iteration are adjustable according to joint identification-verification supervisory signals.

9 An apparatus of claim 8, wherein the joint identification-verification supervisory signals comprises an identification supervisory signal and a verification supervisory signal,

wherein,

the identification supervisory signal is generated by classifying features extracted from an input face region into one of N identities in a training dataset, and taking the

classification error as the supervisory signal, while the verification signal is generated by comparing features extracted from two input face images respectively to tell if they are from the same person, and taking the verification error as the supervisory signal.

10. An apparatus of claim 1, wherein features extracted by a plurality of deep feature extraction hierarchies in the feature extraction unit are concatenated for face recognition.

11. An apparatus of claim 10, wherein distances between the concatenated features extracted from two input face images are compared to a threshold to determine if the two input face images are from the same person for face verification, or distances between features of an input query face image to features of each of face images in a face image database are computed to determine which identity in the face image database the input query face image belongs to for face identification.

12. A method for face recognition, comprising:

configuring a plurality of deep feature extraction hierarchies such that each of the deep feature extraction hierarchies contains a plurality of cascaded convolution layers, local-connection layers, pooling layers and full-connection layers, and neurons in the full-connection layers are only connected to a part of neurons in a previous layer thereof, while neurons in the convolution layers and the local-connection layers are only connected to a part of neurons in local regions in a previous layer thereof;

training the configured deep feature extraction hierarchies to learn neural connections in the convolution layers, the sparse local-connection layers, and the full-connection layers, and to adjust neural weights in these layers;

extracting features from input face images by the trained deep feature extraction hierarchies,; and

recognizing faces based on features extracted from each input face image by the feature extraction unit.

13. A method of claim 12, wherein the training further comprises:

learning iteratively neural connections in the convolution layers, the local-connection layers, and the full-connection layers and neural weights on the neural connections.

14. A method of claim 13, wherein in one iteration, the training comprises:

adjusting the neural weights on neural connections by fixing neural connections in the convolution, local-connection and full-connection layers, and,

pruning the neural connections in one or more of the convolution, local-connection and full-connection layers while fixing the neural weights thereof.

15. A method of claim 14 wherein the neural connections are pruned according to correlations between neural activations of connected neurons, wherein a majority of connections between weakly correlated neurons are pruned while a majority of connections between strongly correlated neurons are reserved.

16. A method of claim 13, wherein before the first iteration, neurons in the full-connection layers are connected to all neurons in a previous layer thereof, while neurons in the sparse convolution modules and the sparse local-connection modules are connected to all neurons in local regions in a previous layer thereof, respectively.

17. A method of claim 13, wherein for the second and the following iterations, the neuron connections are taken from reserved neuron connections in the previous iteration and the neuron weights on these neuron connections are initialized by neuron weights learned in the previous iteration.

18. A method of claim 17, wherein the neuron weights on the reserved neuron connections from the previous iteration are adjustable according to joint identification-verification supervisory signals.

19 A method of claim 18, wherein the joint identification-verification supervisory signals comprises an identification supervisory signal and a verification supervisory signal,

wherein,

the identification supervisory signal is generated by classifying features extracted from an input face region into one of N identities in a training dataset, and taking the classification error as the supervisory signal, while the verification signal is generated by comparing features extracted from two input face images respectively to tell if they are from the same person, and taking the verification error as the supervisory signal.

20. A method of claim 11, wherein features extracted by a plurality of deep feature extraction hierarchies in the feature extraction unit are concatenated for face recognition.

21. A system for face recognition, comprising:

a memory that stores executable components; and

a processor electrically coupled to the memory to execute the executable components to:

configure a plurality of deep feature extraction hierarchies such that each of the deep feature extraction hierarchies contains a plurality of cascaded convolution layers, local-connection layers, pooling layers and full-connection layers, and neurons in the full-connection layers are only connected to a part of neurons in a previous layer thereof, while neurons in the convolution layers and the local-connection layers are only connected to a part of neurons in local regions in a previous layer thereof;

training the configured deep feature extraction hierarchies;

extract features from input face images with the trained deep feature extraction hierarchies; and

recognize faces based on features extracted from each input face image by the feature extraction unit.

22. A system of claim 21, wherein the processor is further configured to execute the executable components for training the configured deep feature extraction hierarchies by adding supervisory signals on them so as to learn neural connections in the convolution layers, the local-connection layers, and the full-connection layers, and to adjust neural weights in these layers.

23. A system of claim 22, wherein the training further comprises:

learning iteratively neural connections in the convolution layers, the local-connection layers, and the full-connection layers and neural weights on the neural connections.

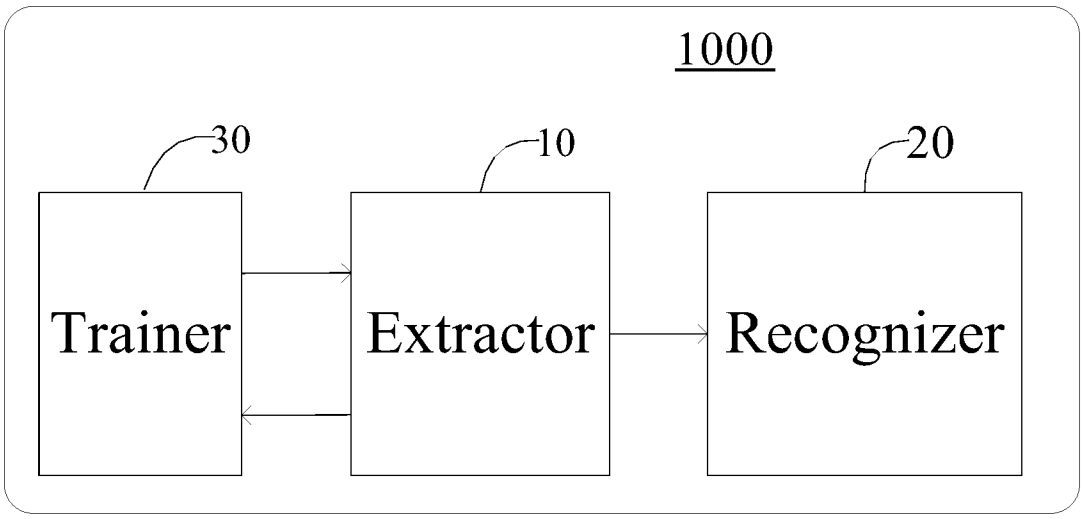
1000
30
10
20
Trainer
Extractor
Recognizer

Fig. 1

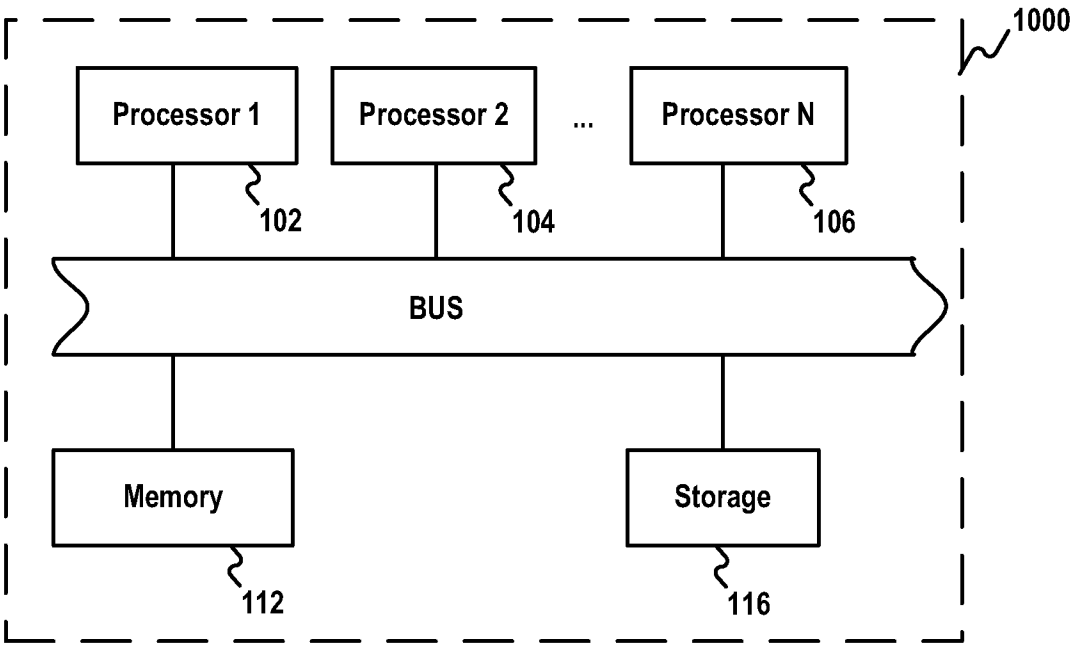
1000
Processor 1
Processor 2
...
Processor N
102
104
106
BUS
Memory
Storage
112
116

Fig. 2

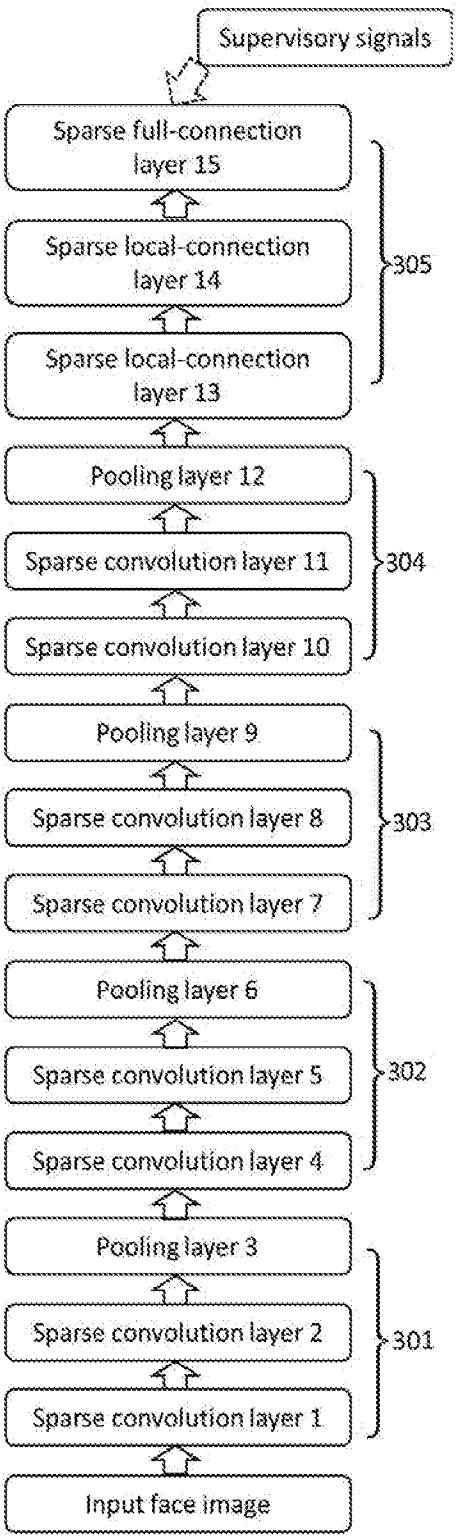
Supervisory signals
Sparse full-connection layer 15
Sparse local-connection layer 14
Sparse local-connection layer 13
305
Pooling layer 12
Sparse convolution layer 11
Sparse convolution layer 10
304
Pooling layer 9
Sparse convolution layer 8
Sparse convolution layer 7
303
Pooling layer 6
Sparse convolution layer 5
Sparse convolution layer 4
302
Pooling layer 3
Sparse convolution layer 2
Sparse convolution layer 1
301
Input face image

Fig. 3

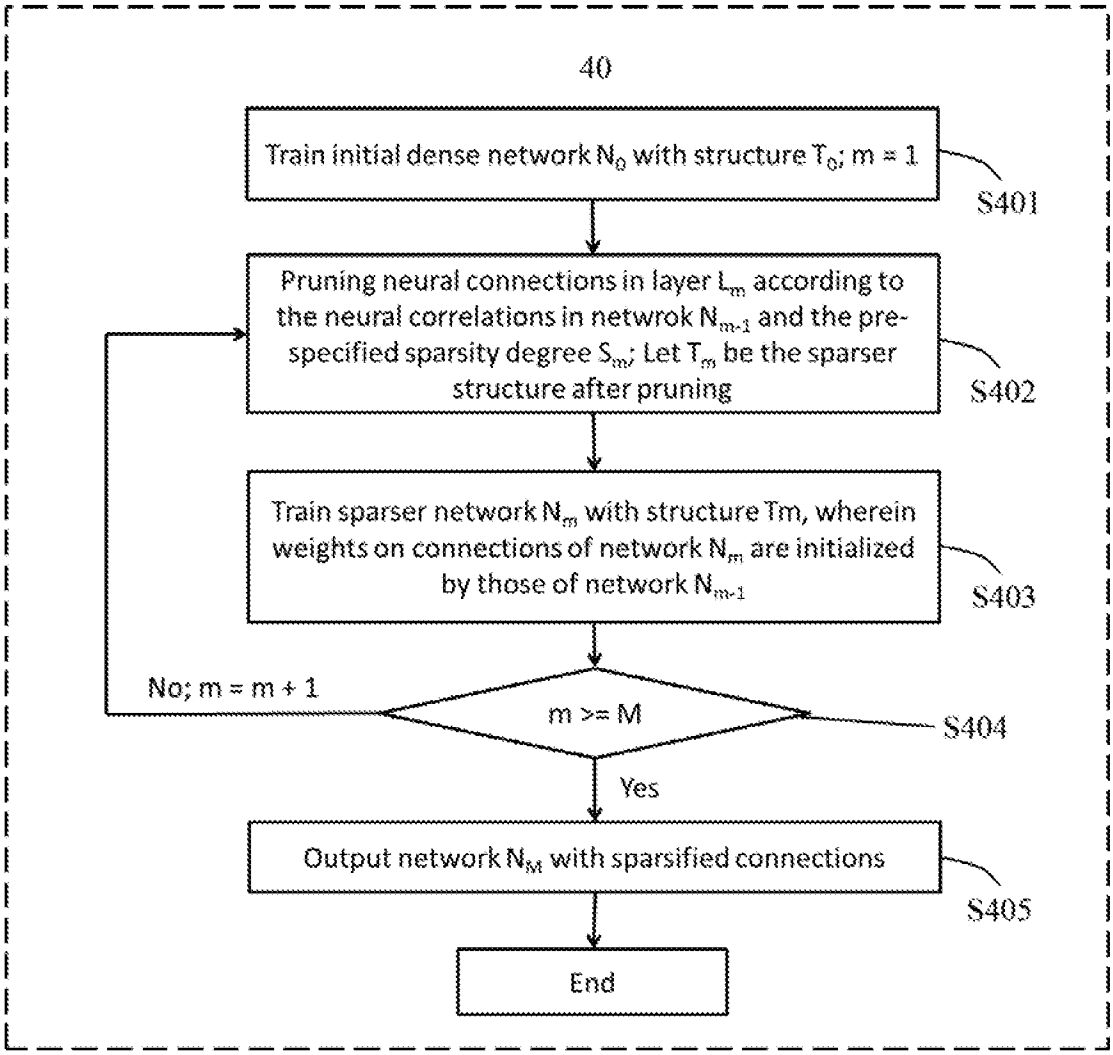
40
Train initial dense network N_0 with structure T_0; m = 1
S401
Pruning neural connections in layer L_m according to the neural correlations in netwrok N_{m-1} and the pre-specified sparsity degree S_m; Let T_m be the sparser structure after pruning
S402
Train sparser network N_m with structure Tm, wherein weights on connections of network N_m are initialized by those of network N_{m-1}
S403
No; m = m + 1
m >= M
S404
Yes
Output network N_M with sparsified connections
S405
End

Fig. 4

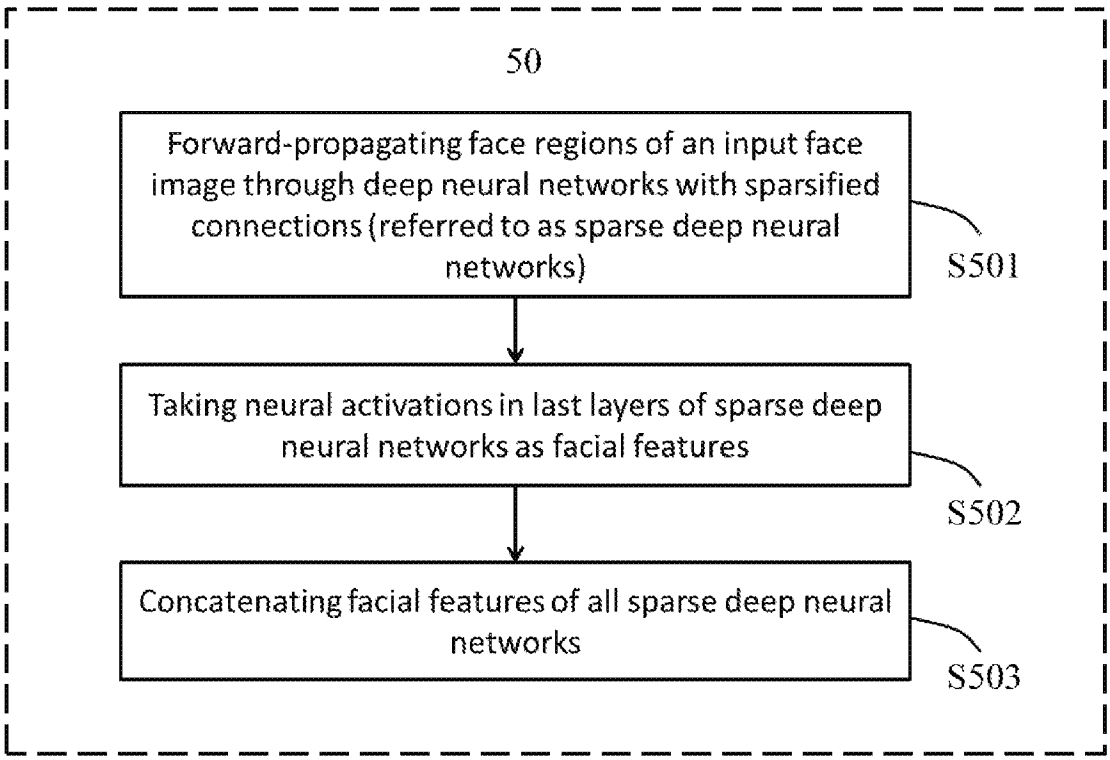
50
Forward-propagating face regions of an input face image through deep neural networks with sparsified connections (referred to as sparse deep neural networks)
S501
Taking neural activations in last layers of sparse deep neural networks as facial features
S502
Concatenating facial features of all sparse deep neural networks
S503

Fig. 5

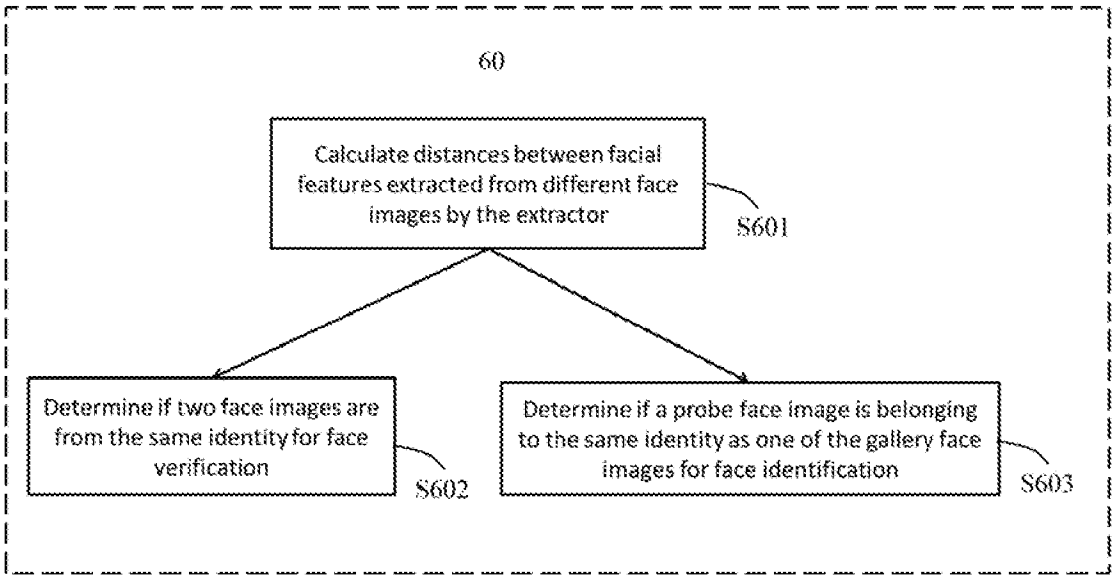
60
Calculate distances between facial features extracted from different face images by the extractor
S601
Determine if two face images are from the same identity for face verification
S602
Determine if a probe face image is belonging to the same identity as one of the gallery face images for face identification
S603

Fig. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2015/093031

A. CLASSIFICATION OF SUBJECT MATTER

G06K 9/00(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06K; G06F; G06T

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

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

CNKI,CNPAT,WPI,EPODOC: feature, face, deep, hierarch+, identity, gallery, convolution, layer, neurons

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2015154206 A1 (TANG, XIAOOU) 15 October 2015 (2015-10-15) description, paragraphs [0007]-[0008], [0029]-[0030]	1-23
A	US 7684651 B2 (MICROSOFT CORPORATION) 23 March 2010 (2010-03-23) the whole document	1-23
A	US 8218880 B2 (MICROSOFT CORPORATION) 10 July 2012 (2012-07-10) the whole document	1-23
A	US 7646894 B2 (MICROSOFT CORPORATION) 12 January 2010 (2010-01-12) the whole document	1-23
A	US 7668346 B2 (MICROSOFT CORPORATION) 23 February 2010 (2010-02-23) the whole document	1-23
A	CN 103530657 A (UNIV. SOUTH CHINA TECHNOLOGY) 22 January 2014 (2014-01-22) the whole document	1-23

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

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
05 May 2016	**28 June 2016**
Name and mailing address of the ISA/CN **STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China** Facsimile No. **(86-10)62019451**	Authorized officer **ZHANG,Xuan** Telephone No. **(86-10)61648236**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2015/093031

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
WO	2015154206	A1	15 October 2015	None			
US	7684651	B2	23 March 2010	US	2008052312	A1	28 February 2008
				US	2010135584	A1	03 June 2010
				US	7860347	B2	28 December 2010
US	8218880	B2	10 July 2012	US	2009297046	A1	03 December 2009
US	7646894	B2	12 January 2010	US	2007189611	A1	16 August 2007
US	7668346	B2	23 February 2010	US	2007223790	A1	27 September 2007
CN	103530657	A	22 January 2014	None			