



(51) International Patent Classification:
G06F 21/32 (2013.01)

(21) International Application Number:
PCT/CN2023/133621

(22) International Filing Date:
23 November 2023 (23.11.2023)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **EBAY INC.** [US/US]; 2025 Hamilton Avenue,
San Jose, California 95125 (US).

(72) Inventor; and

(71) Applicant (for SC only): **XU, Yongsong** [CN/CN]; 2F,
Tower C, German Center Shanghai, No. 88 KeYuan Road,
Pudong New Area, Shanghai 201203 (CN).

(72) Inventors: **ZHU, Jian**; 2F, Tower C, German Center
Shanghai, No. 88 KeYuan Road, Pudong New Area, Shang-
hai 201203 (CN). **TAO, Fujun**; 2F, Tower C, German Cen-
ter Shanghai, No. 88 KeYuan Road, Pudong New Area,
Shanghai 201203 (CN). **WANG, Xiaotong**; 2F, Tower C,
German Center Shanghai, No. 88 KeYuan Road, Pudong
New Area, Shanghai 201203 (CN).

(74) Agent: **CHINA SCIENCE PATENT & TRADEMARK
AGENT LTD.**; Suite 4-312, No. 87, West 3rd Ring North
Rd., Haidian District, Beijing 100089 (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, CV, CZ, DE, DJ, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG,
KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY,
MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA,
NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO,
RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH,
TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS,
ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, CV,
GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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).

(54) Title: IDENTITY VERIFICATION SERVICE AND TECHNIQUE

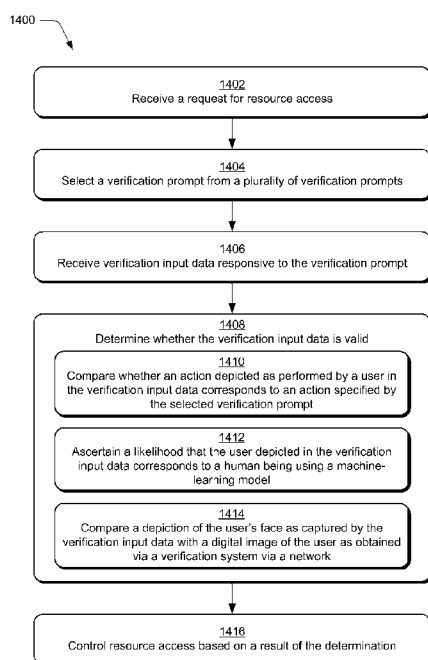


Fig. 14

(57) Abstract: Identity verification services and techniques are described. In one or more implementations, the identity verification service is employed to control resource access based on identity verification. In a first example, an action depicted as performed by a user in verification input data (e.g., head movement, gesture, and so on) is compared by the identity verification service to determine whether the action corresponds to an action specified by a selected verification prompt. In a second example, a likelihood is ascertained by the identity verification service as to whether the user depicted in the verification input data corresponds to a human being through use of a machine-learning model. In a third example, a depiction of the user's face as captured by the verification input data is compared with a digital image of the user maintained by a verification system. Resource access is controlled by the identity verification service.

Published:

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

IDENTITY VERIFICATION SERVICE AND TECHNIQUE

BACKGROUND

[0001] Verification of a user's identity is utilized as a basis by a computing device to control access to a variety of resources, whether for access to resources implemented locally at the computing device and/or remotely as digital services by a service provider system. A variety of techniques and technologies have been developed to implement identity verification, e.g., user passwords, facial recognition, and so on.

[0002] However, these techniques are continually challenged by malicious parties, e.g., through use of photos to detect facial recognition techniques, hacking of camera feeds, employ use of artificial intelligence, and so on. Accordingly, conventional techniques used to identity verification may fail for their intended purpose, thereby exposing these resources to access by malicious parties.

SUMMARY

[0003] Identity verification services and techniques are described. In one or more implementations, the identity verification service is employed to control resource access based on identity verification. In a first example, an action depicted as performed by a user in verification input data (e.g., head movement, gesture, and so on) is compared by the identity verification service to determine whether the action corresponds to an action specified by a selected verification prompt. In a

second example, a likelihood is ascertained by the identity verification service as to whether the user depicted in the verification input data corresponds to a human being through use of a machine-learning model. In a third example, a depiction of the user's face as captured by the verification input data is compared with a digital image of the user maintained by a verification system. Resource access is controlled by the identity verification service based on a result of the determination.

[0004] This Summary introduces a selection of concepts in a simplified form that are further described below in the Detailed Description. As such, this Summary is not intended to identify essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The detailed description is described with reference to the accompanying figures. Entities represented in the figures are indicative of one or more entities and thus reference is made interchangeably to single or plural forms of the entities in the discussion.

[0006] FIG. 1 is an illustration of an environment in an example implementation that is operable to employ identity verification techniques described herein.

[0007] FIG. 2 depicts a system in an example implementation showing training of a machine-learning model of FIG. 1 in greater detail.

[0008] FIG. 3 depicts an example implementation of pseudocode usable to implement training of a machine-learning model.

[0009] FIG. 4 depicts a system in an example implementation in which a prompt module outputs a prompt and verification input data is verified by an action recognition module in response to the prompt for use in identity verification by an identity verification service.

[0010] FIG. 5 depicts a system in an example implementation in which a prompt module and an action recognition module leverage movement as part of an action performed for use in identity verification by an identity verification service.

[0011] FIG. 6 depicts an example implementation showing output of a verification prompt having a movement prompt and generation of a movement response as part of verification input data.

[0012] FIG. 7 depicts an example implementation of pseudocode usable to implement action recognition by the action recognition module.

[0013] FIG. 8 depicts a system in an example implementation in which a prompt module and an action recognition module leverage a gesture as part of identity verification by an identity verification service.

[0014] FIG. 9 depicts an example implementation showing output of a verification prompt having a gesture prompt and generation of a gesture response as part of verification input data.

[0015] FIGS. 10 and 11 depict example implementations of pseudocode usable to implement gesture recognition by the action recognition module.

[0016] FIG. 12 depicts an example implementation showing operation of an ID verification module of FIG. 1 in greater detail.

[0017] FIG. 13 depicts an additional example implementation showing use of identity verification as part of controlling access to a resource configured as an electronic message.

[0018] FIG. 14 is a flow diagram depicting an algorithm as a step-by-step procedure in an example implementation of operations performable for accomplishing a result of identity verification.

[0019] FIG. 15 illustrates an example system including various components of an example device that can be implemented as any type of computing device as described and/or utilize with reference to the previous figures to implement embodiments of the techniques described herein.

DETAILED DESCRIPTION

Overview

[0020] Identity verification of a user is utilized by computing devices to control access to a variety of resources, such as gain local access to a computing device, access digital services remotely via a network, prove access is being attempted by a human being, and so forth. Conventional techniques used to do so, however, are constantly challenged by malicious parties through use of artificial intelligence, use of fake digital images, hacking of a camera feed, and so on.

[0021] Accordingly, identity verification services and techniques are described that are configured to provide additional protection against malicious parties and address technical challenges of conventional identity verification techniques. These techniques are usable to support a variety of computational functionality, examples of which include document verification in order to verify a document upload, user verification to verify a user is a human being, and so forth.

[0022] An identity verification service, for instance, is executable as a digital service at a service provider system, locally at a client device, and so on. The identity verification service is configured to employ a variety of functionalities as part of identity verification. In a first example, a machine-learning model is trained and retrained using positive sample training data and negative sample training data. The positive sample training data includes examples of digital images that capture actual human faces. The negative sample training data, on the other hand, includes examples of digital images that are fake and as such to not

capture actual human faces, e.g., are generated using generative artificial intelligence, are collected from failed access attempts, and so on. The machine-learning model, once trained, is then usable to verify whether a digital image received as part of verification input data is valid, e.g., depicts an actual human being. In this way, the machine-learning model is able to learn and adapt over time as attempts are made by malicious parties that change over time, which is not possible in conventional techniques. Further discussion of training and use of the machine-learning model is described in relation to FIG. 2.

[0023] In a second example, the identity verification system employs a system of prompts and action recognition to determine whether a digital image or other input data (e.g., audio data) depict a corresponding response to a prompt. The prompt, for instance, may be randomly selected from a plurality of prompts that specify actions to be performed by a user, e.g., movement of a user's head, performance of a gesture, a spoken utterance, and so forth. The verification prompt is output in a user interface, and sensors are used to capture a user's response, e.g., using a microphone, camera, and so forth. The verification prompt, for instance, is usable to specify a direction a user is to look, a particular gesture to be performed using the user's hand, and so forth.

[0024] Verification input data generated by the sensors is then processed by an action recognition module to verify whether the verification input data exhibits the action specified by the prompt. The action recognition module, for instance, may compare a target result with a result included in the verification input data, employ

a machine-learning model trained as a classifier to recognize performance of the actions, and so forth. In this way, the identity verification service through the use of prompts is configured to detect whether a user reacts accordingly to perform resource access control. Further discussion of use of prompts and action verification include examples of user movements as described in relation to FIGS. 5-7 and gestures as described in relation to FIGS. 8-11.

[0025] In a third example, a verification system is leveraged that maintains user identities (e.g., drivers licenses, passports, resident identity cards, and so forth) for identity verification. In a first instance, a digital image captured of the user (e.g., during a communication session) is passed along with a legal name of the user to a verification application programming interface (API) of the verification system. The verification system (e.g., through use of resident identity card data) determines whether the digital image depicts the user and returns a result of this determination.

[0026] In a second instance, a query is made to the verification API as before using the user's name, but in this example a digital image of the user is returned, e.g., from a passport, driver's license, employee ID, and so forth. The digital image is then compared with a digital image captured of the user as part of the verification input data to determine by the identity verification service, itself, as to whether the user is valid. In this way, the verification system provides an additional degree of protection against compromise by malicious parties, further discussion of which is described in relation to FIG. 12.

[0027] In the following discussion, an example environment is described that employs the techniques described herein. Example procedures are also described that are performable in the example environment as well as other environments. Consequently, performance of the example procedures is not limited to the example environment and the example environment is not limited to performance of the example procedures.

Example Identity Verification Environment

[0028] FIG. 1 is an illustration of an environment 100 in an example implementation that is operable to employ identity verification techniques described herein. The illustrated environment 100 includes a service provider system 102 and a client device 104 that are communicatively coupled, one to another, via a network 106. Computing devices that implement the service provider system 102 and the client device 104 are configurable in a variety of ways.

[0029] A computing device, for instance, is configurable as a desktop computer, a laptop computer, a mobile device (e.g., assuming a handheld configuration such as a tablet or mobile phone as illustrated), and so forth. Thus, a computing device ranges from full resource devices with substantial memory and processor resources (e.g., personal computers, game consoles) to a low-resource device with limited memory and/or processing resources (e.g., mobile devices). Additionally, although a single computing device is shown, a computing device is also

representative of a plurality of different devices, such as multiple servers utilized by a business to perform operations “over the cloud” as described in FIG. 15.

[0030] The client device 104 includes a communication module 108 that is representative of functionality to communicate via the network 106 with a service manager module 110 of the service provider system 102. The service manager module 110 is configured to implement digital services 112 using hardware and software resources 114, e.g., a processing device and a computer-readable storage medium. Digital services 112 are usable to expose a variety of functionality to the client device 104 via the network through execution by computing devices at the service provider system 102, an example of which is illustrated as an identity verification service 116.

[0031] The identity verification service 116 is configured to employ techniques to verify an identity, such as to verify an identity associated with a document upload, a user “is who they say they are,” verify whether an actual human being is involved in a communication session, and so forth. To do so, the identity verification service 116 is configurable to employ a variety of functionality, examples of which are illustrated as a prompt module 118, an action recognition module 120, and an ID verification module 122. In one or more examples, a machine-learning model 124 is trained to assist in implementation of these techniques as further described in relation to FIG. 2.

[0032] The prompt module 118 is configured to select a verification prompt 126 from a plurality of verification prompts (e.g., randomly), which is communicated

to the client device 104 and output in a user interface as illustrated. Verification input data 128 is then returned in response. The verification input data 128 is processed by the action recognition module 120 to determine whether an action specified in the verification prompt 126 is performed by a user, e.g., head motion, a gesture, and so on. In this way, the identity verification service 116 leverages user responses to the verification prompt 126 to protect against use of fake images (e.g., showing a digital image on a mobile phone), hacking of an image feed, use of artificial intelligence, and so on.

[0033] The client device 104, for instances, includes sensors 130, such as an image capture device 132 (e.g., a camera), an audio capture device 134 (e.g., a microphone), and so on. The client device 104 may also include light emitter devices 136, e.g., as an infrared projector for operation in a relatively dark physical environment, as part of a depth perception technique in which an array of dots is projected to map structure and depth of a human face, and so forth. The verification input data 128 is then generated based on the sensors, e.g., to include digital images, digital audio data, depth maps, and so forth which are utilized to control resource access. Resource access, for instance, is usable to “login” to the client device 104, access digital services 112 of the service provider system 102 (e.g., access a webpage, login to a user account), and so on.

[0034] In another example, an ID verification module 122 is utilized to communicate with a verification system that maintains user identities (e.g., drivers’ licenses, passports, resident identity cards, and so forth) for identity verification.

The ID verification module 122, for instance, passes a digital image captured of a user along with a legal name of the user to a verification application programming interface (API) of the verification system. The verification system (e.g., through use of resident identity card data) determines whether the digital image depicts the user and returns a result of this verification. In a second instance, a query is made to the verification API using the user's name, and a digital image of the user is returned, e.g., from a passport, driver's license, employee ID, and so forth. The digital image is then compared with a digital image captured of the user as part of the verification input data to determine by the identity verification service, itself, as to whether the user is valid.

[0035] In this way, the identity verification service addresses technical challenges of identity verification. Further discussion of training and use of the machine-learning model 124 is described in relation to FIG. 2. Further discussion of use of the prompt module 118 and action recognition module 120 for output of verification prompts and processing verification input data to evaluate actions is described in relation to FIGS. 3-11. Further discussion of implementation of the ID verification module 122 is described in relation to FIG. 12.

[0036] In general, functionality, features, and concepts described in relation to the examples above and below are employed in the context of the example procedures described in this section. Further, functionality, features, and concepts described in relation to different figures and examples in this document are interchangeable among one another and are not limited to implementation in the context of a

particular figure or procedure. Moreover, blocks associated with different representative procedures and corresponding figures herein are applicable together and/or combinable in different ways. Thus, individual functionality, features, and concepts described in relation to different example environments, devices, components, figures, and procedures herein are usable in any suitable combinations and are not limited to the particular combinations represented by the enumerated examples in this description.

Example Identity Verification Techniques

[0037] The following discussion describes identity verification techniques that are implementable utilizing the described systems and devices. Aspects of each of the procedures are implemented in hardware, firmware, software, or a combination thereof. The procedures are shown as a set of blocks that specify operations performable by hardware and are not necessarily limited to the orders shown for performing the operations by the respective blocks. Blocks of the procedures, for instance, specify operations programmable by hardware (e.g., processor, microprocessor, controller, firmware) as instructions thereby creating a special purpose machine for carrying out an algorithm as illustrated by the flow diagram. As a result, the instructions are storable on a computer-readable storage medium that causes the hardware to perform the algorithm.

[0038] FIG. 2 depicts a system 200 in an example implementation showing training of a machine-learning model 124 of FIG. 1 in greater detail. The

machine-learning model 124 is illustrated as being trained using a machine-learning model training system 202. The machine-learning model training system 202 includes a training data generation module 204 that is representative of functionality to generate training data and use the generated training data to train the machine-learning model 124.

[0039] A machine-learning model 124 refers to a computer representation that is tunable (e.g., through training and retraining) based on inputs without being actively programmed by a user to approximate unknown functions, automatically and without user intervention. In particular, the term machine-learning model includes a model that utilizes algorithms to learn from, and make predictions on, known data by analyzing training data to learn and relearn to generate outputs that reflect patterns and attributes of the training data. Examples of machine-learning models include neural networks, convolutional neural networks (CNNs), long short-term memory (LSTM) neural networks, generative adversarial networks (GANs), decision trees, support vector machines, linear regression, logistic regression, Bayesian networks, random forest learning, dimensionality reduction algorithms, boosting algorithms, deep learning neural networks, etc.

[0040] The machine-learning model 124, for instance, is configurable using a plurality of layers having, respectively, a plurality of nodes. The plurality of layers is configurable to include an input layer, an output layer, and one or more hidden layers. Calculations are performed by the nodes within the layers via

hidden states through a system of weighted connections that are “learned” during training of the machine-learning model to implement a variety of tasks.

[0041] In order to train the machine-learning model, training data is generated in this example by a training data generation module 204. The training data generation module 204 includes a positive sample generator module 206 that is configured to generate positive sample training data 210 and negative sample generation module 208 configured to generate negative sample training data 212. The training data provides examples of “what is to be learned” by the machine-learning model 124, i.e., as a basis to learn patterns from the data. The machine-learning model training system 202, for instance, collects and preprocesses the training data that includes input features and corresponding target labels, i.e., of what is exhibited by the input features.

[0042] The machine-learning model training system 202 then utilizes a training module 214 to initialize parameters of the machine-learning model 124. The parameters are used by the machine-learning model 124 as internal variables to represent and process information during training and represent interferences gained through training.

[0043] The training data having the positive sample training data 210 and the negative sample training data 212 is then received as an input by the machine-learning model 124 and used as a basis for generating predictions based on a current state of parameters of layers and corresponding nodes of the model, a result of which is output as output data. Output data describes an outcome of the

task, e.g., as a probability of being a member of a particular class (e.g., human being) in a classification scenario.

[0044] Training of the machine-learning model 124 by the training module 214 includes calculating a loss function 216 to quantify a loss associated with operations performed by nodes of the machine-learning model 124. The calculating of the loss function 216, for instance, includes comparing a difference between predictions specified in the output data with target labels specified by the training data for the positive sample training data 210 and the negative sample training data 212. The loss function 216 is configurable in a variety of ways, examples of which include regret, Quadratic loss function as part of a least squares technique, and so forth.

[0045] Calculation of the loss function 216 also includes use a backpropagation operation as part of minimizing the loss function 216 and thereby training parameters of the machine-learning model 124. Minimizing the loss function 216, for instance, includes adjusting weights of the nodes in order to minimize the loss and thereby optimize performance of the machine-learning model 124 in performance of a particular task. The adjustment is determined by computing a gradient of the loss function 216, which indicates a direction to be used in order to adjust the parameters to minimize the loss. The parameters of the machine-learning model 124 are then updated based on the computed gradient.

[0046] This process continues over a plurality of iterations in an example until a stopping criterion is met. The stopping criterion is employed by the training

module 214 in this example to reduce overfitting of the machine-learning model 124, reduce computational resource consumption, and promote an ability of the machine-learning model 124 to address previously unseen data, i.e., that is not included specifically as an example in the training data. Examples of a stopping criterion include but are not limited to a predefined number of epochs, validation loss stabilization, achievement of a performance improvement threshold, or based on performance metrics such as precision and recall.

[0047] Configuration of the training data is usable to support a variety of usage scenarios. In one example, the training data as described above includes positive sample training data 210 and negative sample training data 212. In an example in which the machine-learning model 124 is trained to recognize actual human faces, the positive sample training data 210 includes digital images captured of actual human faces. The negative sample training data 212, on the other hand, includes fake or “spoofed” faces, e.g., as generated using generative artificial intelligence, as examples received by the identity verification service 116 that have been labeled by a human operator as fake, and so on.

[0048] The machine-learning model 124, once trained, is thus usable to output a probability that a digital image received as part of verification input data 128 corresponds to a “real” human being. Other examples are also contemplated, such as to capture audio data and so forth. FIG. 3 depicts an example implementation 300 of pseudocode usable to implement training of the machine-learning model.

[0049] The machine-learning model 124 is also trainable to support a variety of other identification verification functionality, such as to identify performance of a particular action (e.g., from a plurality of actions as an ensemble model), compare digital images to determine if a human face captured in the digital images correspond to each other, and so on and further described in the following examples.

[0050] FIG. 4 depicts a system 400 in an example implementation in which a prompt module 118 outputs a prompt and verification input data is verified by an action recognition module 120 in response to the prompt for use in identity verification by an identity verification service 116. The prompt module 118, for example, is configured to select a verification prompt 126 from a plurality of prompts 402 maintained in a storage device 404, e.g., randomly. The verification prompt 126 is output and displayed in a user interface by a communication module 108.

[0051] Sensors 130 are then used to capture an action performed by the user in response to the verification prompt 126, which is communicated as verification input data 128 back to the action recognition module 120. The action recognition module 120 then determines whether an action performed in the verification input data 128 corresponds to an action specified by the verification prompt 126. Actions are configurable in a variety of ways, such as movement as described in relation to FIG. 5, a gesture as described in relation to FIG. 8, and so on.

[0052] FIG. 5 depicts a system 500 in an example implementation in which a prompt module 118 and an action recognition module 120 leverage movement as part of an action performed for use in identity verification by an identity verification service 116. As part of identity verification in this example, the identity verification service 116 is configured to determine whether a user responds with actions that correspond with actions requested by the prompt module 118.

[0053] The prompt module 118, of instance, includes a plurality of prompts 502 that are maintained in a storage device 504. The plurality of prompts 502 defines actions that are to be performed by a user as a movement prompt 506 communicated to the client device 104 as part of a verification prompt 126. Verification input data 128 that includes a movement response 508 is then processed by the action recognition module 120 (e.g., using a machine-learning model 124) to determine whether an action performed by the user corresponds with the movement prompt 506 and associated action specified by the verification prompt 126.

[0054] The movement prompt 506 in this instance specifies an action performed using movement of a user's head, e.g., to look up, down, left, right, and so forth. The prompt module 118 is configured to select the movement prompt 506 randomly from the prompts 502 maintained in storage, which is communicated as part of a verification prompt 506 via the network 106 to the client device 104. A

communication module 108 at the client device 104 is then configured to render the movement prompt 506 in a user interface.

[0055] In response, the user is tasked with performing the specified action, the recoding of which is used to generate the movement response 508 as part of the verification input data 128. A light emitter device 136, for instance, is employed to generate a grid of dots (e.g., using infrared wavelengths) onto the user's face. An image capture device 132 then captures a stream of digital images that includes the dots as part of depth estimation to uniquely identify the user and subsequent movements of the user, which is included as part of the movement response 508 of the verification input data 128.

[0056] FIG. 6 depicts an example implementation 600 showing output of a verification prompt 126 having a movement prompt 506 and generation of a movement response 508 as part of verification input data 128. This example implementation is illustrated using a first stage 602 and a second stage 604 depicting rendering of the verification prompt 126 in a user interface.

[0057] At the first stage 602, the verification prompt 126 includes a visual guide 606 configured to guide alignment of a depiction of a head 608 of the user in a digital image as captured by an image capture device of the client device. The visual guide 606, for instance, is configured to provide real time feedback in the user interface based on one or more digital images captured of the user. In this way, the verification prompt 126 is configured to ensure that a subject of the action (e.g., the user's head) is positioned within view as part of digital images

captured by the image capture device 132 of the client device 104. The verification prompt 126 also includes text 610 describing an action to be performed, i.e., the movement prompt 506, which in this example specifies that the user is to “look to their left.”

[0058] At the second stage 602, the verification prompt 126 includes output of a representation of a timer indicative of an amount of time available to capture a digital image of the user as performing the action, examples of which include a text countdown 612 and a graphical countdown 614. The timer is utilized to limit an amount of time, in which, the verification input data 128 may be captured, which helps to protect against compromise by malicious parties, e.g., by providing a plurality of movements in an attempt that one “matches” using recorded digital video.

[0059] The user interface at the second stage 604 also provides a real time output of digital images as providing feedback to the user regarding the action performed. The sensors 130 (e.g., the image capture device 132) are then utilized by the communication module 108 to generate the verification input data 128 which is used as a basis for identity verification by the action recognition module 120. FIG. 7 depicts an example implementation 700 of pseudocode usable to implement action recognition by the action recognition module 120. The action recognition module 120, for example, is configurable to employ object detection, machine learning by the machine-learning model 124, and so on to determine whether the movement response 508 corresponds to the action specified by the movement

prompt 506. Based on this determination, the identity verification service 116 is usable to control resource access, e.g., to verify a user's identity, verification the user is a human being, document upload, and so forth.

[0060] FIG. 8 depicts a system 800 in an example implementation in which a prompt module 118 and an action recognition module 120 leverage a gesture as part of identity verification by an identity verification service 116. As part of identity verification in this example, the identity verification service 116 is configured to determine whether a user responds with a gesture that correspond with a performance of a gesture requested by the prompt module 118.

[0061] The prompt module 118, of instance, includes a plurality of prompts 502 that are maintained in a storage device 504 as previously described. The plurality of prompts 502 defines gestures that are to be performed by a user as a gesture prompt 802 communicated to the client device 104 as part of a verification prompt 126. Verification input data 128 that includes a gesture response 804 is then processed by the action recognition module 120 (e.g., using a machine-learning model 124) to determine whether a gesture performed by the user corresponds with the gesture prompt 802.

[0062] The gesture prompt 802 in this instance specifies an action performed using movement of a user's head, e.g., to look up, down, left, right, and so forth. The prompt module 118 is configured to select the gesture prompt 802 randomly from the prompts 502 maintained in storage, which is communicated as part of a verification prompt 506 via the network 106 to the client device 104. A

communication module 108 at the client device 104 is then configured to render the gesture prompt 802 in a user interface.

[0063] In response, the user is tasked with performing the specified gesture, the recording of which is used to generate the gesture response 804 as part of the verification input data 128. The client device 104, for instance, may utilize sensors 130 (e.g., an image capture device 132, touchscreen input, trackpad functionality, and so on), a light emitter device 136, and so on to capture performance of the gesture, e.g., using a plurality of digital images streamed as a digital video. The action recognition module 120 is then employed to determine whether the gesture response 804 corresponds with the gesture prompt 802, e.g., the gesture is performed by the user.

[0064] FIG. 9 depicts an example implementation 900 showing output of a verification prompt 126 having a gesture prompt and generation of a gesture response as part of verification input data 128. This example implementation is illustrated using a first stage 902 and a second stage 904 depicting rendering of the verification prompt 126 in a user interface.

[0065] At the first stage 902, the verification prompt 126 includes a graphical depiction 906 of a hand as an illustration of a gesture to be performed as part of the action, e.g., to raise a single finger and thumb in the illustrated example. The verification prompt 126 also includes a textual description 908 of the gesture. Other examples are also contemplated, including audio output that describes the gesture.

[0066] The verification prompt 126 also includes a visual guide 910 to guide placement of the user's hand (e.g., depicted as a border specifying which hand to use) with respect to an image capture device 132, which does so using real time feedback through output of a plurality of images.

[0067] At the second stage 902, the verification prompt 126 includes output of a representation of a timer indicative of an amount of time available to capture a digital image of the user as performing the action, examples of which include a text countdown 912 and a graphical countdown 914. The timer is utilized to limit an amount of time, in which, the verification input data 128 may be captured, which helps to protect against compromise by malicious parties, e.g., by providing a plurality of movements in an attempt that one "matches" using recorded digital video.

[0068] The user interface, like the user interface of FIG. 6, provides a real time output of digital images as providing feedback to the user regarding the gesture 916 performed. The sensors 130 are then utilized by the communication module 108 to generate the verification input data 128 which is used as a basis for identity verification by the action recognition module 120. FIGS. 10 and 11 depict example implementations 1000, 1100 of pseudocode usable to implement gesture recognition by the action recognition module 120. The action recognition module 120, for example, is configurable to employ object detection, machine learning by the machine-learning model 124, and so on to determine whether the gesture response 804 corresponds to the gesture specified by the gesture prompt 802.

Based on this determination, the identity verification service 116 is also usable to control resource access, e.g., to verify a user's identity, verification the user is a human being, document upload, and so forth.

[0069] FIG. 12 depicts an example implementation 1200 showing operation of the ID verification module 122 of FIG. 1 in greater detail. The ID verification module 122 is configured to communicate with a verification application programming interface (illustrated as verification API 1202) of a verification system 1204. The verification includes a storage device 1206 configured to maintain identity documents associated with a user, e.g., resident identity card, employee badge, driver's license, passport, and so on. The identity documents 1208 are usable by the ID verification module 122 to verify a user's identity in a variety of ways.

[0070] In a first instance, a communication module 108 generates the verification input data 128 to include a user name 1212 (e.g., legal name) of the user along with digital image 1214 captured by the image capture device 132 of the user. The communication module 108, for instance, captures the digital image 1214 during a communication session with the user at a point-in-time, at which, the user is unaware that the digital image is captured. The communication module 108, for instance, may make the user aware that the digital image is going to be captured and gain permission for such capture, but then capture the image at a random point in time during the communication session to protect against compromise by malicious parties. The verification system 1204 (e.g., through use of resident

identity card data) determines whether the digital image 1214 depicts the user corresponding to the user name 1212 and returns a result of this determination.

[0071] In a second instance, a query is made to the verification API 1202 by the ID verification module 122 as previously described that includes the user name 1212. In this example, however, a digital image 1210 of the user is returned by the verification system 1204, e.g., from a passport, driver's license, employee ID, and so forth. The digital image 1210 from the verification system 1204 is then compared with a digital image 1214 captured of the user as part of the verification input data 128 to determine by the identity verification service 116, itself, as to whether the user is valid. In this way, the identity verification service 116 provides an additional degree of protection against compromise by malicious parties.

[0072] FIG. 13 depicts an additional example implementation 1300 showing use of identity verification as part of controlling access to a resource configured as an electronic message. In this example, a gesture 1302 is selected from a plurality of prompts 302 to control access to a body 1304 of an email message to be sent to a user 1306. Accordingly, upon receipt of the message a verification prompt 126 is output in a user interface of the gesture in this example. Verification input data 128 generated in response is evaluated through execution of the identity verification service 116, e.g., as executable instructions included as part of the message, through communicative via the network 106 to the service provider system 102, and so forth. In this way, access to the computing resource as the

digital message (e.g., email, direct message, and so forth) is controlled by a sender of the message using the techniques described herein.

[0073] FIG. 14 is a flow diagram depicting an algorithm 1400 as a step-by-step procedure in an example implementation of operations performable for accomplishing a result of identity verification. To begin in this example, a request is received for resource access (block 1402), e.g., to access a digital service 112, a client device 104, a particular communication module 108 (e.g., application), verify whether the user is a human being, as part of a document upload, and so forth.

[0074] A verification prompt 126 is selected from a plurality of verification prompts (block 1404), e.g., prompts 302. A prompt module 118, for instance, selects the verification prompt 126 from a plurality of prompts 302 as examples of movement prompts, gesture prompts, and other types of prompts. Verification input data 128 is received responsive to the verification prompt 126 (block 1406) at the identity verification service 116.

[0075] A determination is then made by the identity verification service 116 as to whether the verification input data 128 is valid (block 1408). In a first example, an action depicted as performed by a user in the verification input data is compared by an action recognition module 120 to determine whether the action corresponds to an action specified by the selected verification prompt (block 1410). In a second example, a likelihood is ascertained as to whether the user depicted in the verification input data corresponds to a human being using a machine-learning

model (block 1412), e.g., such as a machine-learning model 124 trained as described in relation to FIG. 2. In a third example, a depiction of the user's face as captured by the verification input data 128 is compared with a digital image of the user as obtained via a verification system via a network (block 1414) by an ID verification module 122. Resource access is controlled by the identity verification service 116 based on a result of the determination (block 1416). A variety of other examples are also contemplated.

Example System and Device

[0076] FIG. 15 illustrates an example system generally at 1500 that includes an example computing device 1502 that is representative of one or more computing systems and/or devices that implement the various techniques described herein. This is illustrated through inclusion of the identity verification service 116. The computing device 1502 is configurable, for example, as a server of a service provider, a device associated with a client (e.g., a client device), an on-chip system, and/or any other suitable computing device or computing system.

[0077] The example computing device 1502 as illustrated includes a processing device 1504, one or more computer-readable media 1506, and one or more I/O interface 1508 that are communicatively coupled, one to another. Although not shown, the computing device 1502 further includes a system bus or other data and command transfer system that couples the various components, one to another. A system bus can include any one or combination of different bus structures, such as

a memory bus or memory controller, a peripheral bus, a universal serial bus, and/or a processor or local bus that utilizes any of a variety of bus architectures. A variety of other examples are also contemplated, such as control and data lines.

[0078] The processing device 1504 is representative of functionality to perform one or more operations using hardware. Accordingly, the processing device 1504 is illustrated as including hardware element 1510 that is configurable as processors, functional blocks, and so forth. This includes implementation in hardware as an application specific integrated circuit or other logic device formed using one or more semiconductors. The hardware elements 1510 are not limited by the materials from which they are formed or the processing mechanisms employed therein. For example, processors are configurable as semiconductor(s) and/or transistors (e.g., electronic integrated circuits (ICs)). In such a context, processor-executable instructions are electronically-executable instructions.

[0079] The computer-readable storage media 1506 is illustrated as including memory/storage 1512 that stores instructions that are executable to cause the processing device 1504 to perform operations. The memory/storage 1512 represents memory/storage capacity associated with one or more computer-readable media. The memory/storage 1512 includes volatile media (such as random access memory (RAM)) and/or nonvolatile media (such as read only memory (ROM), Flash memory, optical disks, magnetic disks, and so forth). The memory/storage 1512 includes fixed media (e.g., RAM, ROM, a fixed hard drive, and so on) as well as removable media (e.g., Flash memory, a removable hard

drive, an optical disc, and so forth). The computer-readable media 1506 is configurable in a variety of other ways as further described below.

[0080] Input/output interface(s) 1508 are representative of functionality to allow a user to enter commands and information to computing device 1502, and also allow information to be presented to the user and/or other components or devices using various input/output devices. Examples of input devices include a keyboard, a cursor control device (e.g., a mouse), a microphone, a scanner, touch functionality (e.g., capacitive or other sensors that are configured to detect physical touch), a camera (e.g., employing visible or non-visible wavelengths such as infrared frequencies to recognize movement as gestures that do not involve touch), and so forth. Examples of output devices include a display device (e.g., a monitor or projector), speakers, a printer, a network card, tactile-response device, and so forth. Thus, the computing device 1502 is configurable in a variety of ways as further described below to support user interaction.

[0081] Various techniques are described herein in the general context of software, hardware elements, or program modules. Generally, such modules include routines, programs, objects, elements, components, data structures, and so forth that perform particular tasks or implement particular abstract data types. The terms “module,” “functionality,” and “component” as used herein generally represent software, firmware, hardware, or a combination thereof. The features of the techniques described herein are platform-independent, meaning that the

techniques are configurable on a variety of commercial computing platforms having a variety of processors.

[0082] An implementation of the described modules and techniques is stored on or transmitted across some form of computer-readable media. The computer-readable media includes a variety of media that is accessed by the computing device 1502. By way of example, and not limitation, computer-readable media includes “computer-readable storage media” and “computer-readable signal media.”

[0083] “Computer-readable storage media” refers to media and/or devices that enable persistent and/or non-transitory storage of information (e.g., instructions are stored thereon that are executable by a processing device) in contrast to mere signal transmission, carrier waves, or signals per se. Thus, computer-readable storage media refers to non-signal bearing media. The computer-readable storage media includes hardware such as volatile and non-volatile, removable and non-removable media and/or storage devices implemented in a method or technology suitable for storage of information such as computer readable instructions, data structures, program modules, logic elements/circuits, or other data. Examples of computer-readable storage media include but are not limited to RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, hard disks, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or other storage device,

tangible media, or article of manufacture suitable to store the desired information and are accessible by a computer.

[0084] “Computer-readable signal media” refers to a signal-bearing medium that is configured to transmit instructions to the hardware of the computing device 1502, such as via a network. Signal media typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as carrier waves, data signals, or other transport mechanism. Signal media also include any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media.

[0085] As previously described, hardware elements 1510 and computer-readable media 1506 are representative of modules, programmable device logic and/or fixed device logic implemented in a hardware form that are employed in some embodiments to implement at least some aspects of the techniques described herein, such as to perform one or more instructions. Hardware includes components of an integrated circuit or on-chip system, an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), a complex programmable logic device (CPLD), and other implementations in silicon or other hardware. In this context, hardware operates as a processing device that performs

program tasks defined by instructions and/or logic embodied by the hardware as well as a hardware utilized to store instructions for execution, e.g., the computer-readable storage media described previously.

[0086] Combinations of the foregoing are also be employed to implement various techniques described herein. Accordingly, software, hardware, or executable modules are implemented as one or more instructions and/or logic embodied on some form of computer-readable storage media and/or by one or more hardware elements 1510. The computing device 1502 is configured to implement particular instructions and/or functions corresponding to the software and/or hardware modules. Accordingly, implementation of a module that is executable by the computing device 1502 as software is achieved at least partially in hardware, e.g., through use of computer-readable storage media and/or hardware elements 1510 of the processing device 1504. The instructions and/or functions are executable/operable by one or more articles of manufacture (for example, one or more computing devices 1502 and/or processing devices 1504) to implement techniques, modules, and examples described herein.

[0087] The techniques described herein are supported by various configurations of the computing device 1502 and are not limited to the specific examples of the techniques described herein. This functionality is also implementable all or in part through use of a distributed system, such as over a “cloud” 1514 via a platform 1516 as described below.

[0088] The cloud 1514 includes and/or is representative of a platform 1516 for resources 1518. The platform 1516 abstracts underlying functionality of hardware (e.g., servers) and software resources of the cloud 1514. The resources 1518 include applications and/or data that can be utilized while computer processing is executed on servers that are remote from the computing device 1502. Resources 1518 can also include services provided over the Internet and/or through a subscriber network, such as a cellular or Wi-Fi network.

[0089] The platform 1516 abstracts resources and functions to connect the computing device 1502 with other computing devices. The platform 1516 also serves to abstract scaling of resources to provide a corresponding level of scale to encountered demand for the resources 1518 that are implemented via the platform 1516. Accordingly, in an interconnected device embodiment, implementation of functionality described herein is distributable throughout the system 1500. For example, the functionality is implementable in part on the computing device 1502 as well as via the platform 1516 that abstracts the functionality of the cloud 1514.

[0090] In implementations, the platform 1516 employs a “machine-learning model” that is configured to implement the techniques described herein. A machine-learning model refers to a computer representation that can be tuned (e.g., trained and retrained) based on inputs to approximate unknown functions. In particular, the term machine-learning model can include a model that utilizes algorithms to learn from, and make predictions on, known data by analyzing training data to learn and relearn to generate outputs that reflect patterns and attributes of the

training data. Examples of machine-learning models include neural networks, convolutional neural networks (CNNs), long short-term memory (LSTM) neural networks, decision trees, and so forth.

[0091] Although the invention has been described in language specific to structural features and/or methodological acts, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as example forms of implementing the claimed invention.

CLAIMS

What is claimed is:

1. A method comprising:
 - selecting, by a processing device, a verification prompt from a plurality of verification prompts;
 - receiving, by the processing device, verification input data responsive to the verification prompt;
 - determining, by the processing device, whether the verification input data is valid, the determining including:
 - ascertaining whether an action depicted as performed by a user in the verification input data corresponds to an action specified by the selected verification prompt; and
 - ascertaining a likelihood that the user depicted in the verification input data corresponds to a human being using a machine-learning model;
 - and
 - controlling, by the processing device, resource access based on a result of the determining.
2. The method as described in claim 1, wherein the verification prompt specifies a movement prompt to be performed using movement of a head of the user.

3. The method as described in claim 1, wherein the verification prompt specifies a movement prompt to be performed using movement of a hand of the user.

4. The method as described in claim 3, wherein the movement of the hand involves a gesture.

5. The method as described in claim 1, wherein the verification prompt is configured for display in a user interface at a client device associated with the user.

6. The method as described in claim 5, wherein the verification prompt includes a visual guide configured to guide alignment of a depiction of the user in a digital image as captured by an image capture device of the client device.

7. The method as described in claim 5, wherein the verification prompt includes a visual guide configured to provide real time feedback in the user interface based on one or more digital images captured of the user.

8. The method as described in claim 5, wherein the verification prompt includes output of a representation of a timer indicative of an amount of time available to capture a digital image of the user as performing the action.

9. The method as described in claim 5, wherein the verification prompt includes a graphical depiction of the action to be performed.

10. The method as described in claim 9, wherein the graphical depiction depicts a hand as an illustration of a gesture to be performed as part of the action.

11. The method as described in claim 5, wherein the verification prompt includes a visual guide configured to guide alignment of a depiction of a hand of the user in a digital image as captured by an image capture device of the client device.

12. The method as described in claim 1, wherein the determining includes comparing a depiction of a face of the user as captured by the verification input data with a digital image of the user as obtained via a verification system via a network.

13. The method as described in claim 12, wherein the digital image is included as part of a passport or a driver's license.

14. The method as described in claim 1, wherein the machine-learning model is trained using positive sample training data having real user depictions and negative sample training data having fake user depictions.

15. The method as described in claim 14, wherein the fake user depictions are generated using generative artificial intelligence as implemented using machine learning.

16. One or more computer-readable storage media storing instructions that, responsive to execution by a processing device, causes the processing device to perform operations including:

- selecting a verification prompt from a plurality of verification prompts;
- receiving verification input data responsive to the verification prompt;
- controlling resource access by determining whether the verification input data is valid, the determining including:

- ascertaining whether an action depicted as performed by a user in the verification input data corresponds to an action specified by the selected verification prompted; and

- ascertaining a likelihood that the user depicted in the verification input data corresponds to a human being using a machine-learning model.

17. The one or more computer-readable storage media as described in claim 16, wherein the action specifies movement of a hand of the user or a head of the user.

18. A computing device comprising:
an image capture device;
a processing device; and
a computer readable storage media storing instructions that, responsive to execution by the processing device, causes the processing device to perform operations including:

displaying a verification prompt in a user interface, the verification prompt specifying an action to be performed by a user;

displaying a plurality of digital images in real time as captured by the image capture device in the user interface, the plurality of digital images depicting the user; and

receiving access to a resource responsive to a determination made that the plurality of digital images depict the user as performing the action.

19. The computing device as described in claim 18, wherein the action specifies movement of a hand of the user or a head of the user.

20. The computing device as described in claim 18, wherein the determining includes comparing a depiction in a digital image of a face of the user as captured by the image capture device at a point-in-time that is not indicated in the user interface with a digital image obtained from a verification system.

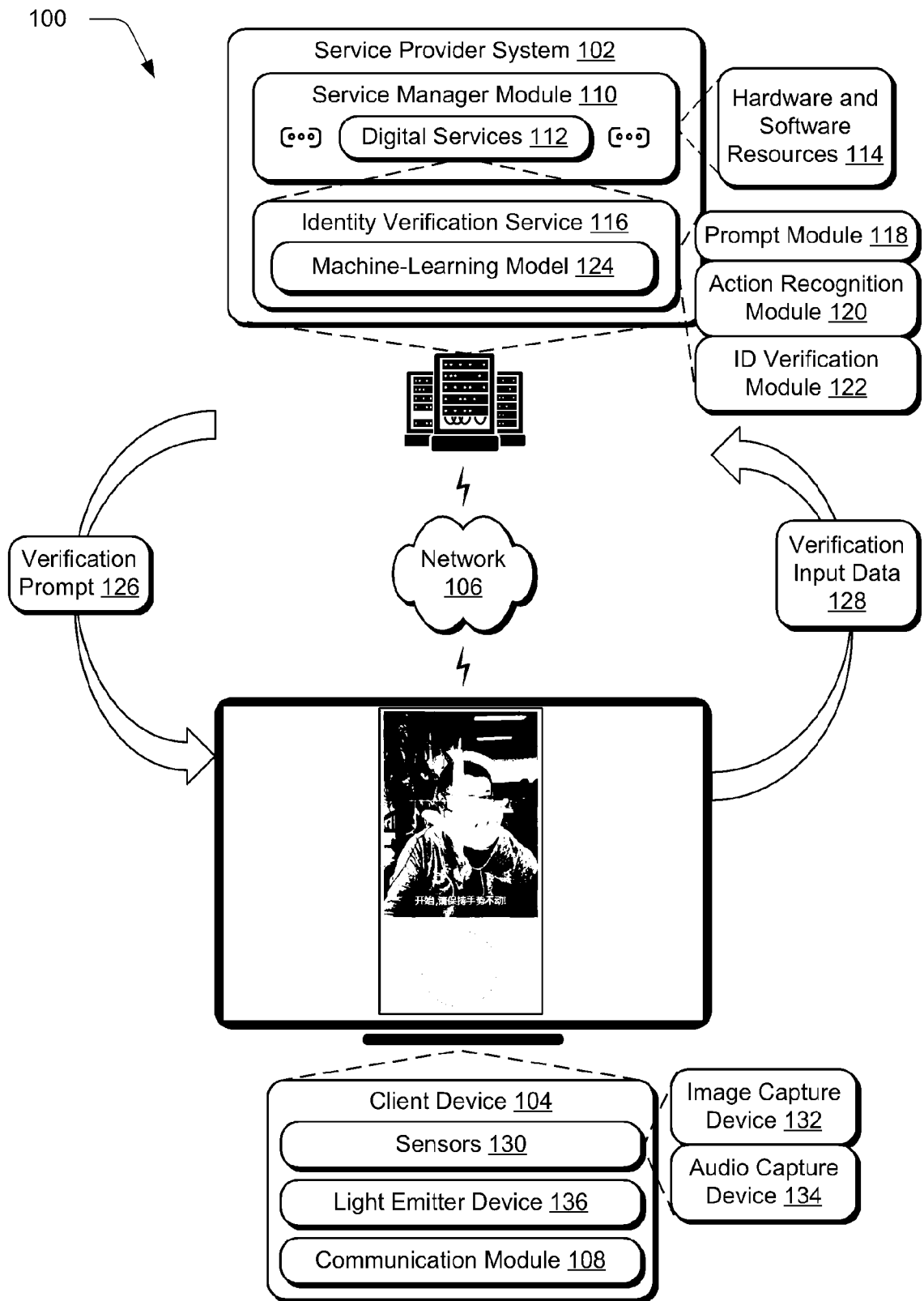


Fig. 1

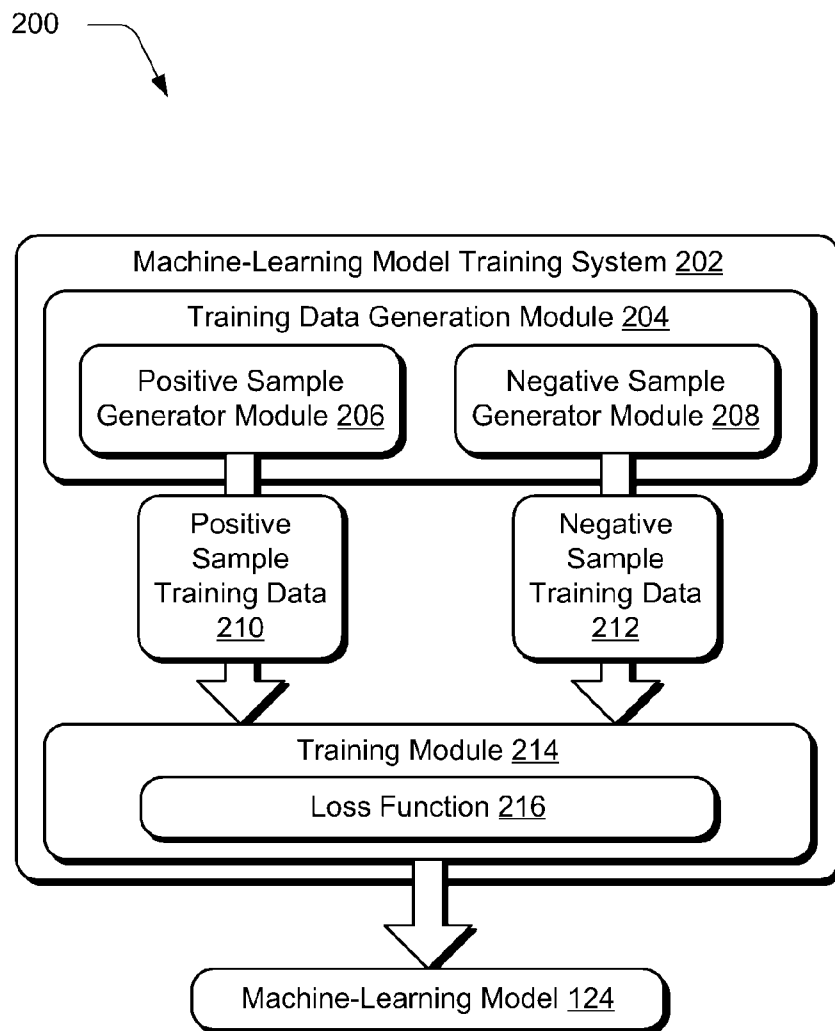


Fig. 2

300



```

class LivenessNet:
    @staticmethod
    def build(width, height, depth, classes):
        # initialize the model along with the input shape to be
        # "channels last" and the channels dimension itself
        model = Sequential()
        inputShape = (height, width, depth)
        chanDim = -1

        # if we are using "channels first", update the input shape
        # and channels dimension
        if K.image_data_format() == "channels_first":
            inputShape = (depth, height, width)
            chanDim = 1

        # first CONV => RELU => CONV => RELU => POOL layer set
        model.add(Conv2D(16, (3, 3), padding="same",
            input_shape=inputShape))
        model.add(Activation("relu"))
        model.add(BatchNormalization(axis=chanDim))
        model.add(Conv2D(16, (3, 3), padding="same"))
        model.add(Activation("relu"))
        model.add(BatchNormalization(axis=chanDim))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))

        # second CONV => RELU => CONV => RELU => POOL layer set
        model.add(Conv2D(32, (3, 3), padding="same"))
        model.add(Activation("relu"))
        model.add(BatchNormalization(axis=chanDim))
        model.add(Conv2D(32, (3, 3), padding="same"))
        model.add(Activation("relu"))
        model.add(BatchNormalization(axis=chanDim))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))

        # first (and only) set of FC => RELU layers
        model.add(Flatten())
        model.add(Dense(64))
        model.add(Activation("relu"))
        model.add(BatchNormalization())
        model.add(Dropout(0.5))

        # softmax classifier
        model.add(Dense(classes))
        model.add(Activation("softmax"))

        # return the constructed network architecture
        return model

```

Fig. 3

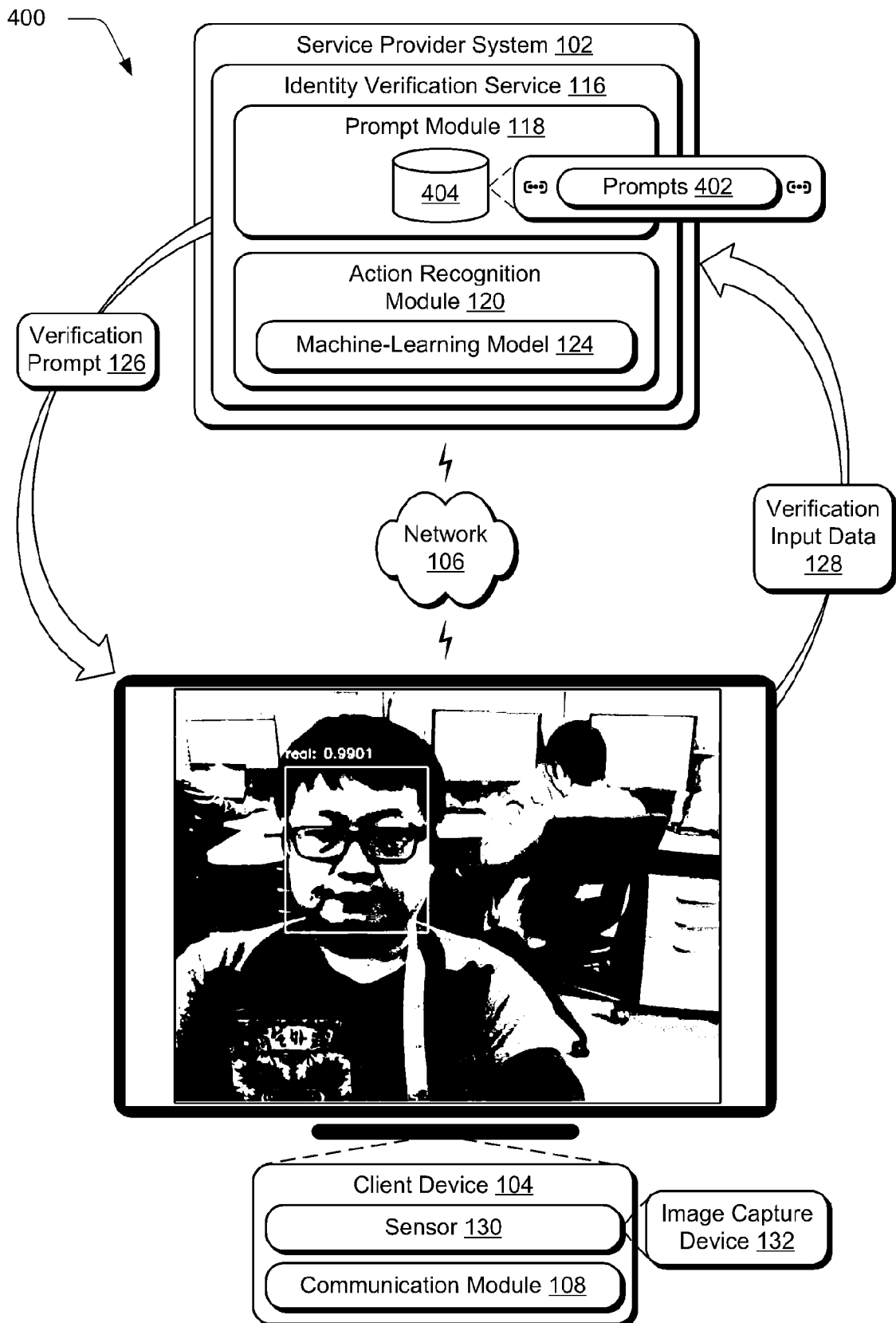


Fig. 4

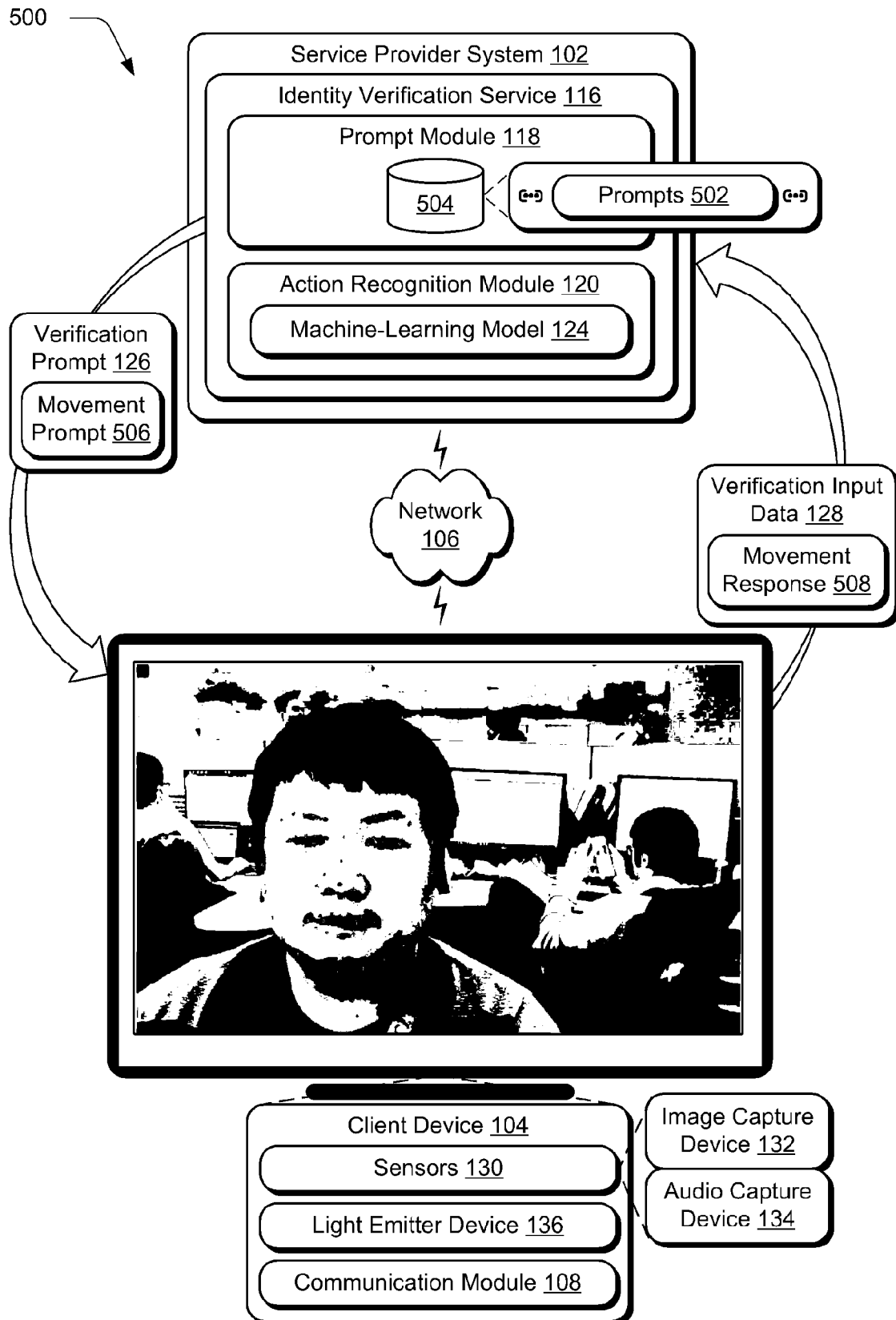


Fig. 5

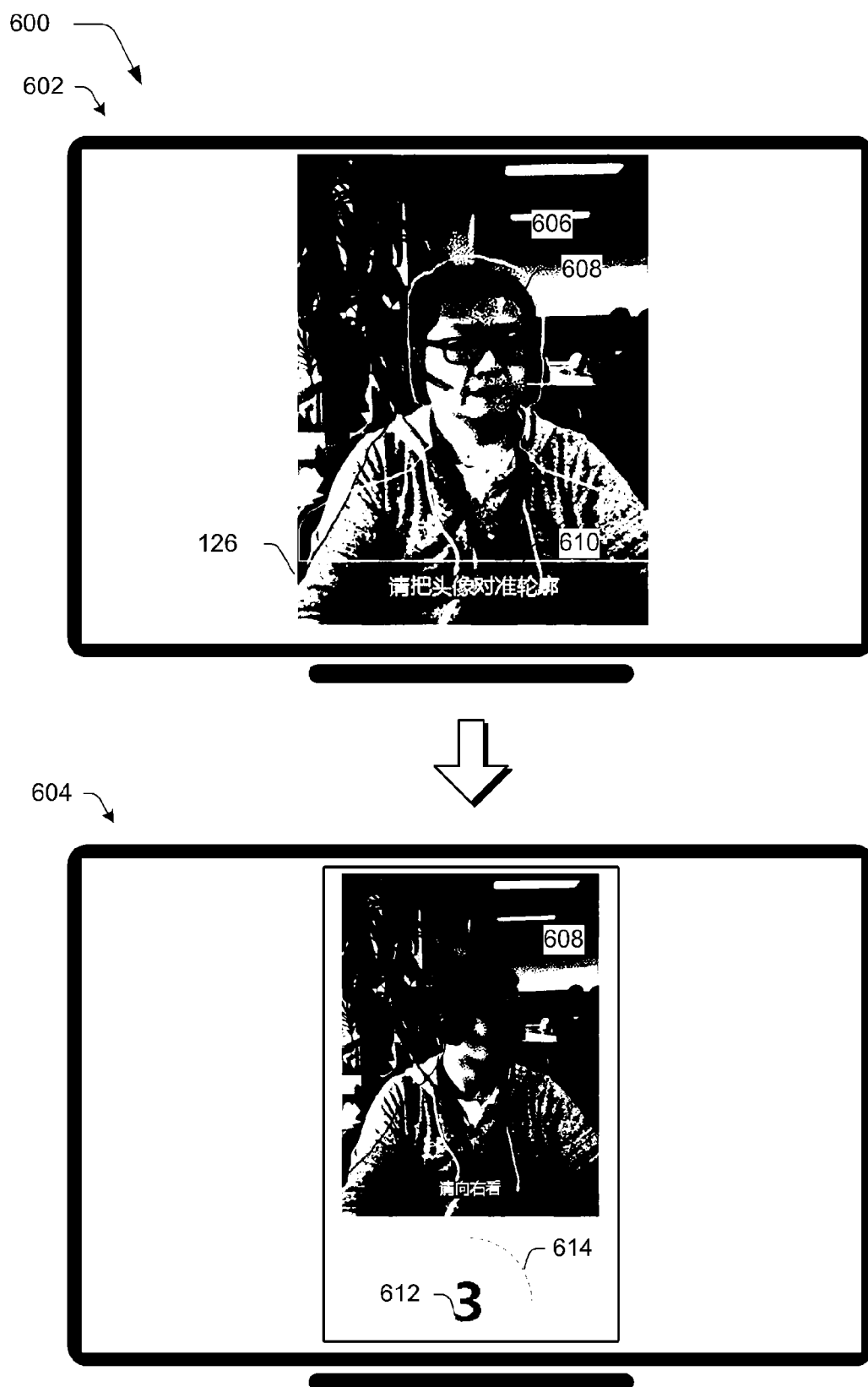


Fig. 6

700

```

if(!_this.is_alive_head==true){
    if(!_this.last_time==0 || (new Date().getTime()-_this.last_time>500 && new Date().getTime()-_this.
last_time<10000 ) ){
        var xdiff_left = positions[62][0] - positions[2][0] ;
        var ydiff_left = positions[62][1] - positions[2][1] ;
        var dis_left = Math.pow((xdiff_left * xdiff_left + ydiff_left * ydiff_left), 0.5);
        var xdiff_right = positions[12][0] - positions[62][0] ;
        var ydiff_right = positions[12][1] - positions[62][1] ;
        var dis_right = Math.pow((xdiff_right * xdiff_right + ydiff_right * ydiff_right), 0.5);
        var xdiff_side = positions[12][0] - positions[2][0] ;
        var ydiff_side = positions[12][1] - positions[2][1] ;
        var dis_side = Math.pow((xdiff_side * xdiff_side + ydiff_side * ydiff_side), 0.5);
        var dis_left_right = Math.abs(dis_left - dis_right);
        //
        //
        document.getElementById('result').innerHTML = dis_left_right;
        document.getElementById('result').innerHTML = dis_side/2){
            _this.aliveStatus=true
            _this.is_head_ok=true;
            _this.is_alive_head=false;
        }
        _this.last_dis_left_right=dis_left_right;
        _this.last_time = new Date().getTime();
    }
}

```

Fig. 7

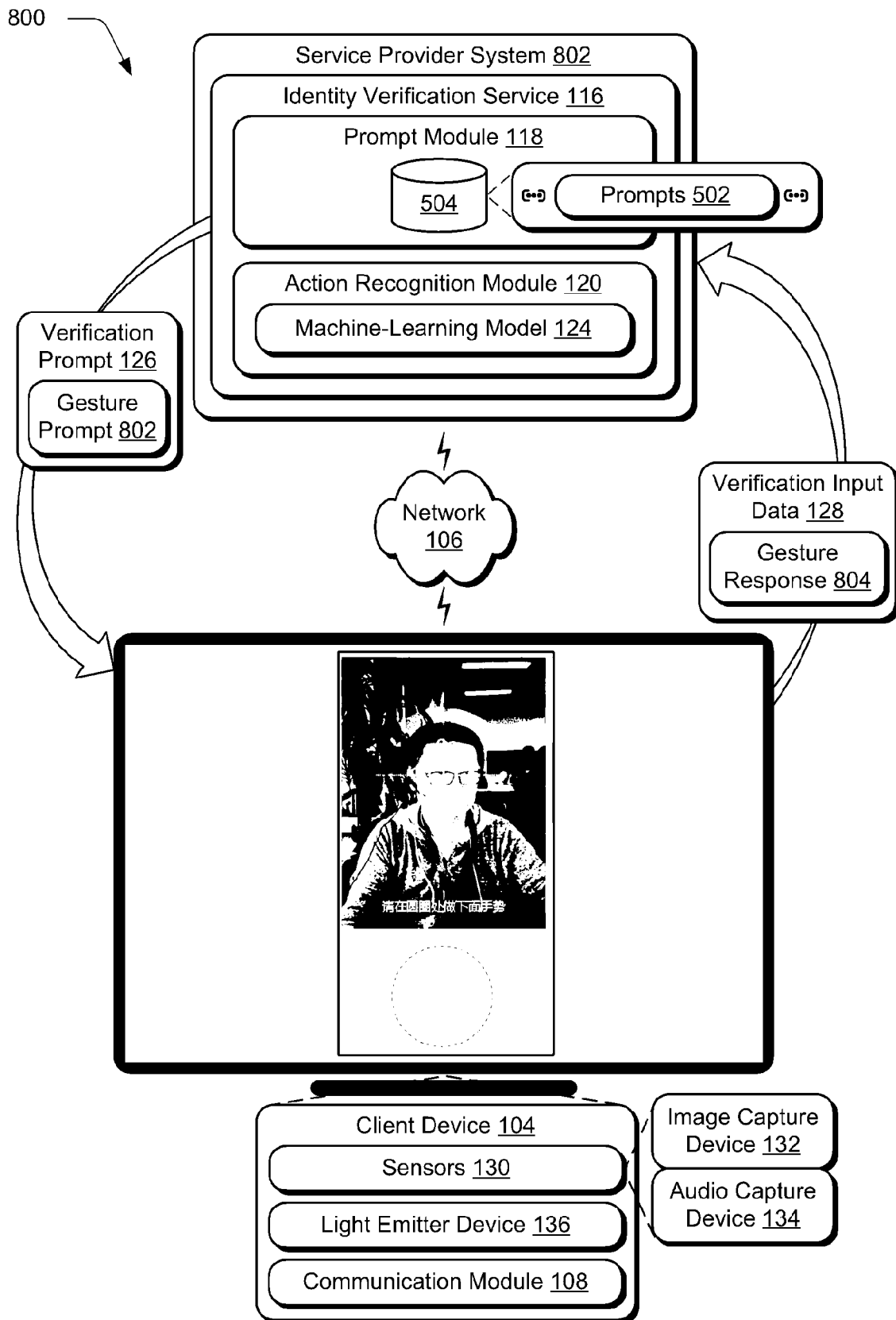


Fig. 8

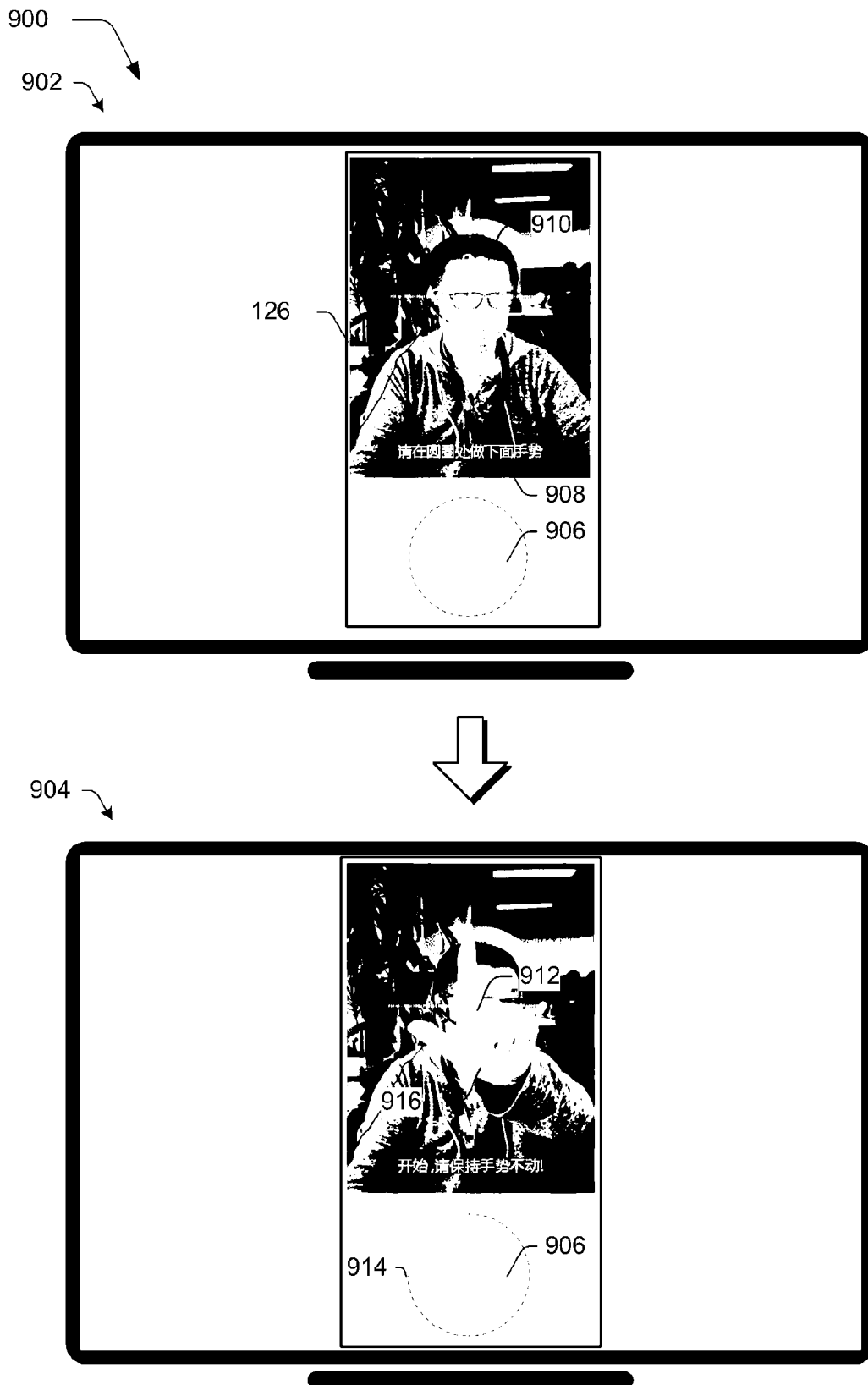


Fig. 9

1000 

```

class handDetector():
    def __init__(self, mode=False, maxHands=2,modelComplexity=1, detectionCon=0.8, trackCon=0.5):
        self.mode = mode
        self.maxHands = maxHands
        self.detectionCon = detectionCon
        self.modelComplex = modelComplexity
        self.trackCon = trackCon
        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(self.mode, self.maxHands,self.modelComplex,self.detectionCon, self.trackCon)
        self.mpDraw = mp.solutions.drawing_utils

    def findHands(self, img, draw=True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) #
        self.results = self.hands.process(imgRGB) #

        if self.results.multi_hand_landmarks: #
            for handLms in self.results.multi_hand_landmarks:
                if draw: #
                    self.mpDraw.draw_landmarks(img, handLms, self.mpHands.HAND_CONNECTIONS)

        return img

    def findPosition(self, img, handNo=0, draw=True):
        lmList = [] #
        if self.results.multi_hand_landmarks:
            myHand = self.results.multi_hand_landmarks[handNo]
            for idNum, lm in enumerate(myHand.landmark):
                h, w, c = img.shape #
                cx, cy = int(lm.x * w), int(lm.y * h) #
                lmList.append([idNum, cx, cy])
                #cv2.circle(img, (cx, cy), 15, (0, 0, 255), cv2.FILLED)
            if draw: #
                #cv2.circle(img, (cx, cy), 15, (0, 0, 255), cv2.FILLED)
                cv2.putText(img, "x"+str(idNum), (cx, cy), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 1)

        return lmList

```

Fig. 10

1100



```
def get_findHands(lmList):
    output = "" #
    tipIds = [4, 8, 12, 16, 20] #

    if len(lmList) != 0: #
        fingers = []

        #
        # 43
        if lmList[tipIds[3]][1] < lmList[tipIds[0]][1] :
            if lmList[tipIds[0]][1] > lmList[tipIds[0] - 1][1]:
                fingers.append(1)
            else:
                fingers.append(0)
        else:
            if lmList[tipIds[0]][1] < lmList[tipIds[0] - 1][1]:
                fingers.append(1)
            else:
                fingers.append(0)

        #
        for idNum in range(1, 5):
            if lmList[tipIds[idNum]][2] < lmList[tipIds[idNum] - 2][2]:
                fingers.append(1)
            else:
                fingers.append(0)
        print(fingers)

    if fingers[0] == 1: #
        output=output + "1"
        #output += 1
    if fingers[1] == 1: #
        output=output + "2"
        #output += 2
    if fingers[2] == 1: #
        output=output + "3"
        #output += 4
    if fingers[3] == 1: #
        output=output + "4"
        #output += 8
    if fingers[4] == 1: #
        output=output + "5"
        #output += 16
    return output
```

Fig. 11

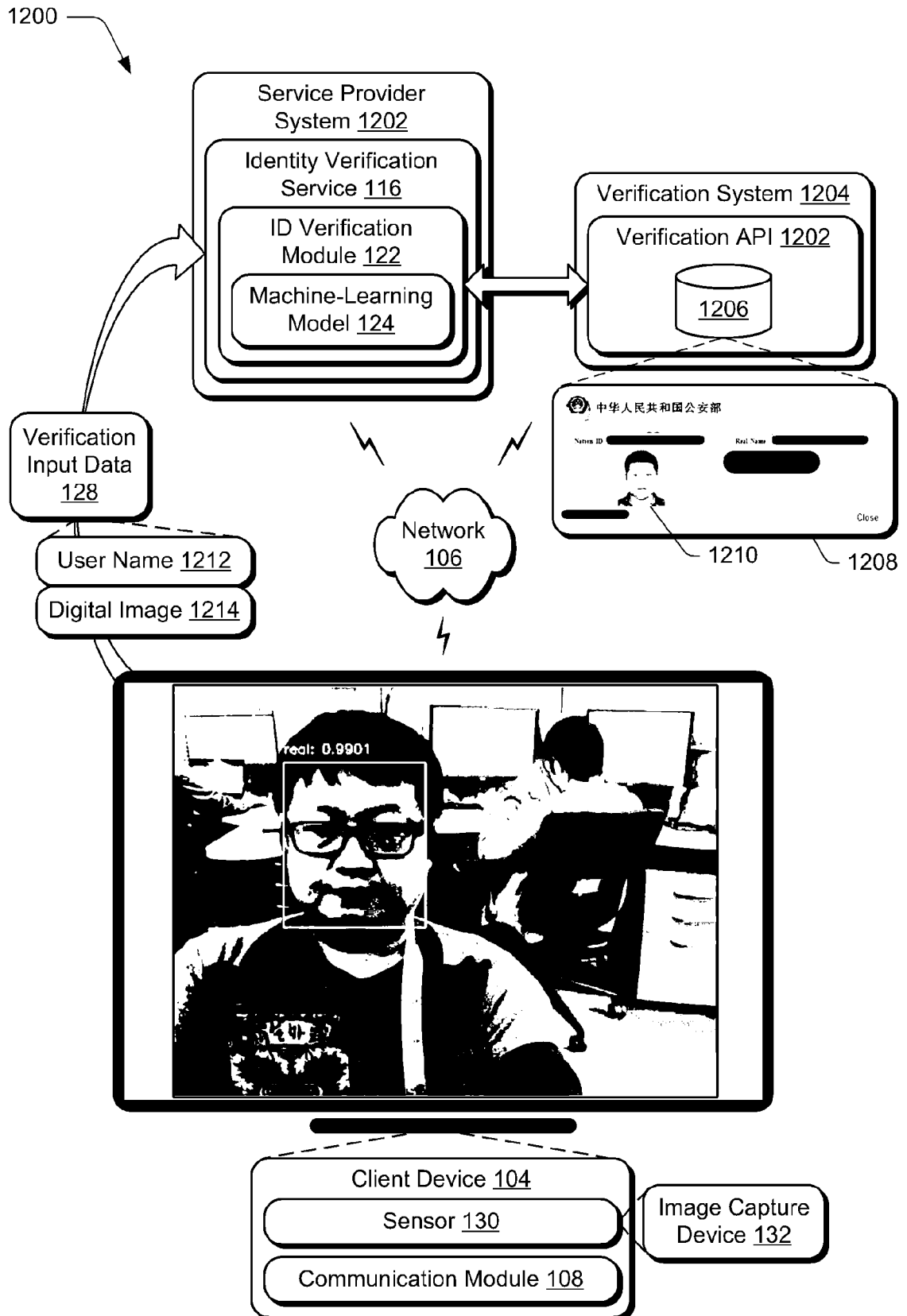


Fig. 12

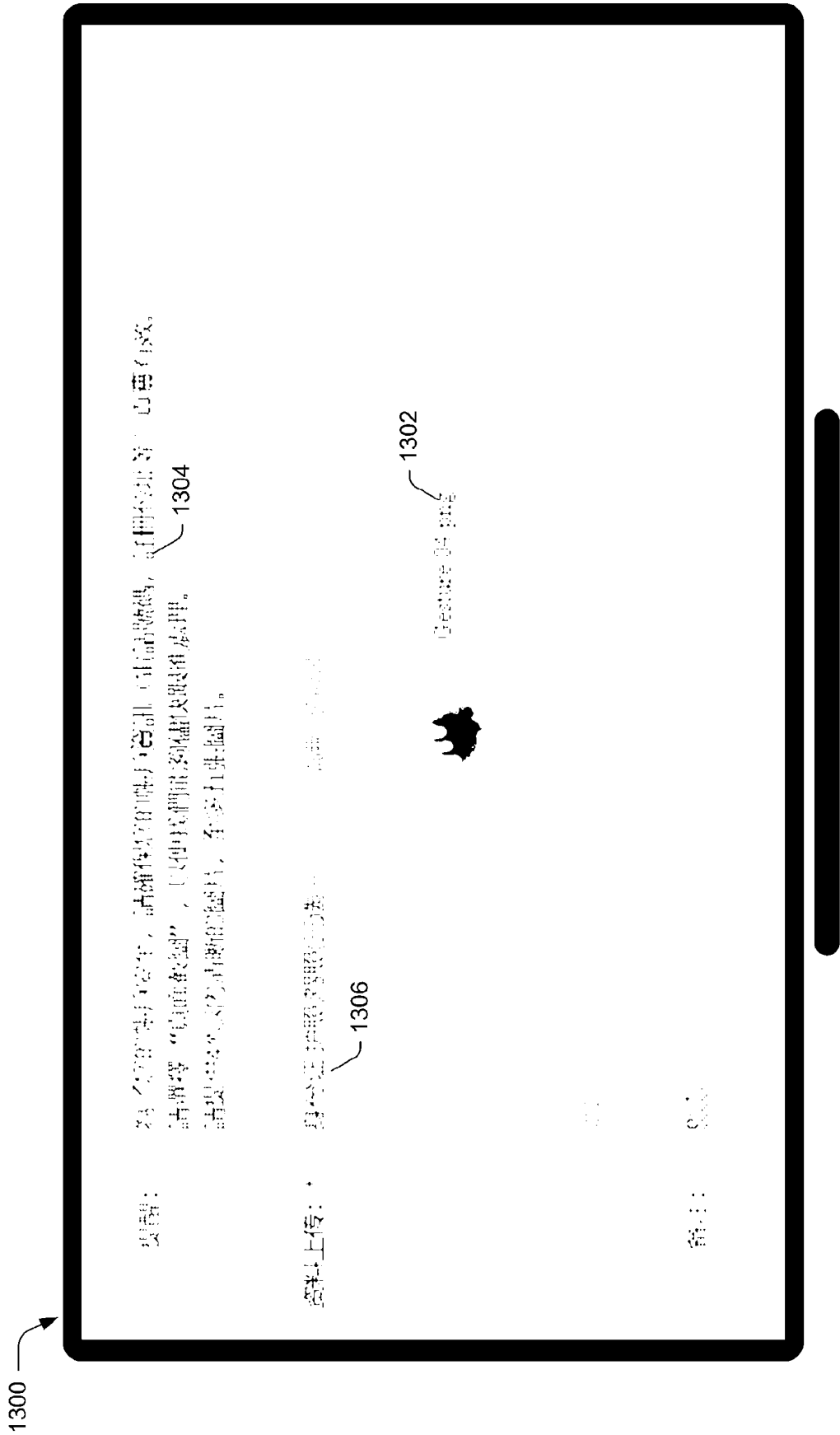


Fig. 13

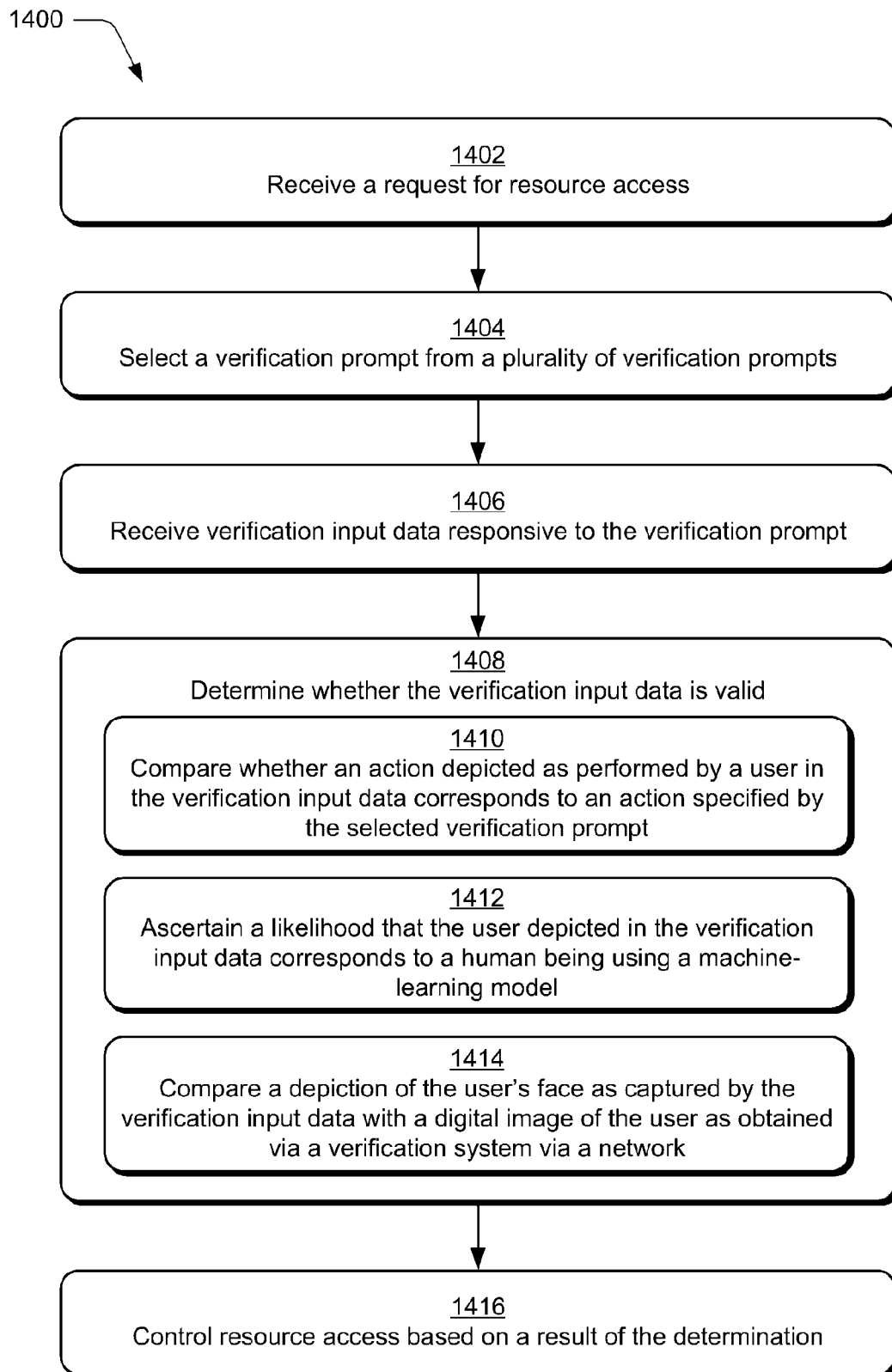


Fig. 14

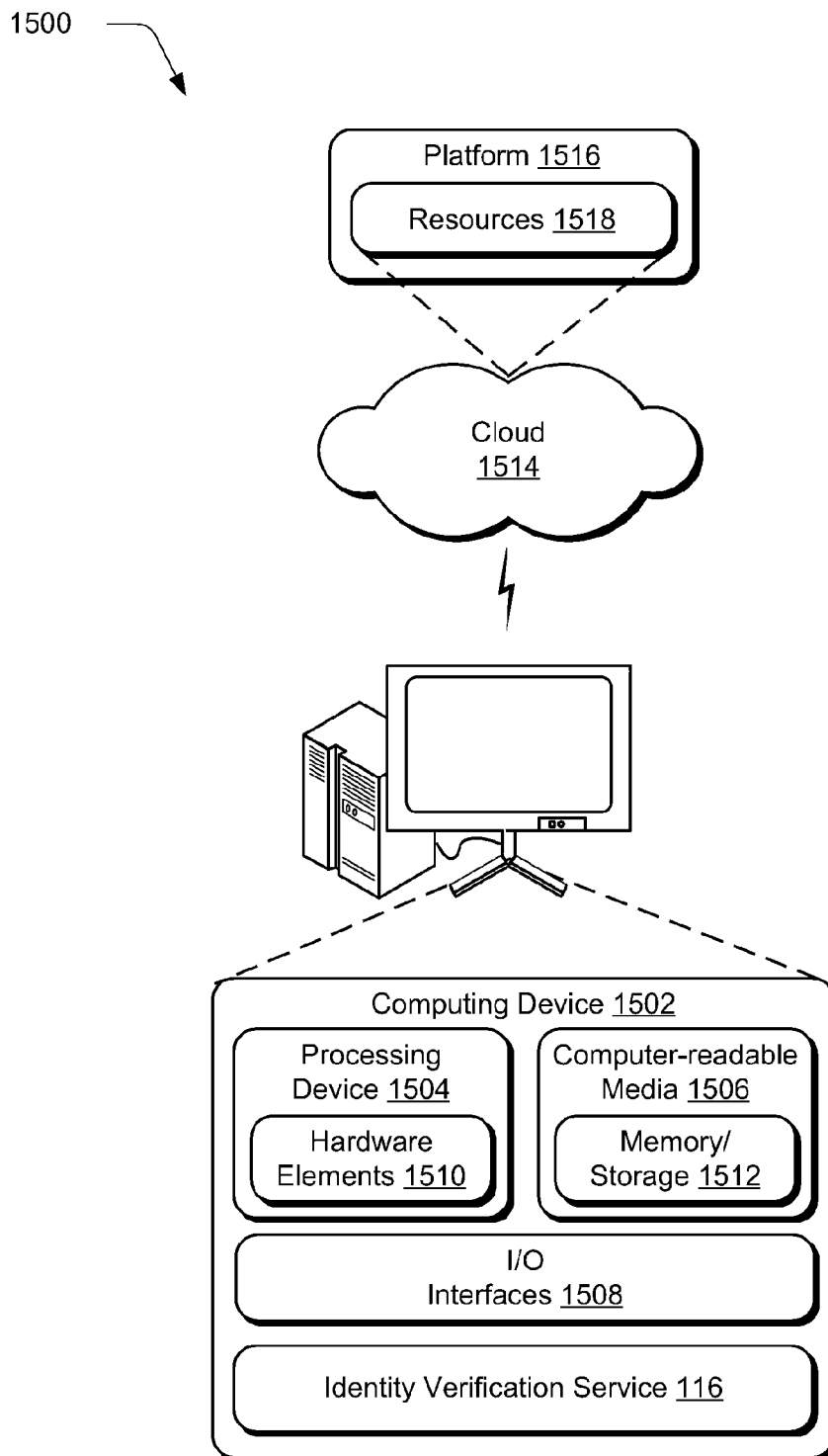


Fig. 15

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2023/133621

A. CLASSIFICATION OF SUBJECT MATTER G06F 21/32(2013.01)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) IPC:G06F 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) CNTXT、ENTXT、ENTXTC、DWPI、CNKI: verification, prompt, ascertain, action, human, resource, access		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 106921621 A (ALIBABA GROUP HOLDING LTD) 04 July 2017 (2017-07-04) claims 1-21, paragraphs [0075]-[0176],abstract	1-20
A	CN 102624705 A (SouthWest Petroleum University) 01 August 2012 (2012-08-01) the whole document	1-20
A	CN 104135365 A (ABAB) ALIBABA GROUP HOLDING LTD05 November 2014 (2014-11-05) the whole document	1-20
A	US 2021342430 A1 (CAPITAL ONE SERVICES LLC) 04 November 2021 (2021-11-04) the whole document	1-20
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: “A” document defining the general state of the art which is not considered to be of particular relevance “D” document cited by the applicant in the international application “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 21 March 2024		Date of mailing of the international search report 08 April 2024
Name and mailing address of the ISA/CN CHINA NATIONAL INTELLECTUAL PROPERTY ADMINISTRATION 6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China		Authorized officer MING,Mei Telephone No. (+86) 62411851

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2023/133621

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN	106921621	A	04 July 2017	None	
CN	102624705	A	01 August 2012	None	
CN	104135365	A	05 November 2014	None	
US	2021342430	A1	04 November 2021	None	