November 2021

Image and/or Video Transformation Using Interpretable Transformation Parameters

Igor Ramos

Daniel Klein

IMAGE AND/OR VIDEO TRANSFORMATION USING INTERPRETABLE TRANSFORMATION PARAMETERS

FIELD

The present disclosure relates generally to using machine-learned model(s) to perform image transformations according to interpretable parameters, which may, in some instances be user-specified or controllable. As one example, a distortion effect can be added or removed from imagery based on user input such as, for example, lens prescription information which can in some cases be inferred from the image.

BACKGROUND

Image transformation has generally been performed manually using image manipulation software. As an example, an image depicting an environment in the summer can be transformed to depict the same environment in the winter using image manipulation techniques. Recently, machine-learned models (e.g., neural networks, etc.) have been trained and used to perform these image transformations.

However, these image transformation models are generally incapable of providing fine-grained control of specific interpretable characteristics of the image that is to be transformed. As an example, such image transformation models are generally incapable of controlling the degree to which a transformation is applied. Further, such image transformation models are generally only capable of providing image transformations of a single type.

One example scenario where image transformation may be desirable is when a user is virtually trying on glasses frames while shopping online. Depending on the shape of the frame and the user’s prescription, the level of distortion may vary and a user may wish to see the distortion effect prior to purchasing a glasses frame. As a further example, a user may desire to remove a distortion effect when in a videoconference while wearing prescription glasses.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 depicts a block diagram of an example computing system according to example embodiments of the present disclosure

Figure 2 depicts a block diagram of an example computing device according to example embodiments of the present disclosure.

Figure 3 depicts a block diagram of an example computing device according to example embodiments of the present disclosure.

Figure 4 depicts a graphical diagram of example machine-learned models for output image generation and associated training signals according to example embodiments of the present disclosure.

Figure 5 depicts a graphical diagram of example machine-learned models for reconstructive input image generation and associated training signals according to example embodiments of the present disclosure.

Figure 6 depicts a graphical representation of an example facial distortion transformation applied to a user image according to example embodiments of the present disclosure.

DETAILED DESCRIPTION

Overview

Example embodiments of the present disclosure are directed to performing transformations of images using machine-learned model(s) and interpretable user-specified parameters that control the transformations. As one example, a distortion effect can be added or removed from imagery based on user input such as, for example, lens prescription information.

More specifically, a machine-learned model can be used to perform transformation(s) on an image based on different parameter(s) of a user-specified conditioning vector. The user can specify desired values for defined characteristics of a transformation (e.g., area of transformation, degree of transformation, etc.) for one or more transformations of the defined characteristic(s) (e.g., transforming lack of facial displacement distortion to presence of a facial displacement distortion, transforming glasses frames to lack of glasses frames, unshaded to shaded, etc.). The user-desired values can be parameterized in the user-specified conditioning vector. As an example, a user may wish to transform an image that depicts a user wearing planar glasses frames to distorted prescription glasses frames. The user-specified conditioning vector can include parameters that indicate the type(s) of transformation(s) the user desires (e.g., prescription, etc.) and/or the degree of transformation desired (e.g., full distortion, partial distortion, etc.).

The machine learned model(s) can receive the input image and the user-specified conditioning vector and transform the input image based on the user’s desired values for the defined characteristics of an output image. Specifically, the output image can correspond to the input image transformed to have the desired values for the defined characteristic(s). Thus, in the example given above, if the user-specified conditioning vector indicates a 60% transformation of the input image that depicts the planar glasses frames, the output image can depict the same image but transformed 60% into the facial distortion according to the prescription. In such fashion, the machine-learned model can perform one or more transformations on an input image based on user-desired values for the defined characteristics of the transformation(s). Stated differently, a user can be provided with continuous fine-grained control of specific, interpretable characteristics of an image that is transformed.

In some implementations, the computing system can obtain at least one user image. For example, the computing system can obtain the at least one user image by a camera associated with the computing system. As another example, the computing system can obtain the at least one user image by a user uploading an image to the computing system. In particular, the at least one user image can include a user image where the user can be wearing glasses frames.

In some implementations, the computing system can obtain personalized data. In particular, the personalized data can be related to a glasses prescription. For example, the user can input information specifying user glasses information. As another example, the computing system can determine personalized data relating to glasses prescription information based at least in part on the user image. For example, the computing system can determine the glasses prescription information based on facial distortion present in the user image. In particular, determining the glasses prescription information based on facial distortion present in the user image can be based at least in part on the shape of the glasses frames. Furthermore, the user image can include the user wearing glasses frames with a prescription.

In some implementations, the personalized data can include other data indicative of a glasses prescription. For example, the personalized data can include glasses lens optical characteristics. Specifically, the personalized data can include correction factors such as cylinder, sphere, axis, or prism data, inter-pupillary distance, etc.

In some implementations, the computing system can provide at least one user image and glasses prescription to a machine-learning network. For example, the machine-learning network can be a generative adversarial network.

In some implementations, the computing system can generate a modified user image. As another example, the computing system can add glasses frames to the user image. In particular, the modified user image can include a user wearing glasses frames where the glasses frames are determined by user input. For example, the user can select a particular glasses frame from a predetermined selection. As another example, the user can upload an image of glasses frames (e.g., a personal image, an image found on the internet, glasses frames on a person, glasses frames not on a person, etc.).

In some implementations, the computing system can determine glasses frames to add to a user image based on historical user data. For example, the computing system can use historical video conference data where the user was wearing glasses. Even more particularly, the computing system can add to a user image the same glasses frames as were previously present in user images.

In one specific use case, the computing system can add glasses frames along with facial distortion when a user is shopping for glasses frames virtually. The user may desire to see the degree of facial distortion a particular glasses frames based on a user's personal glasses prescription. In particular, facial distortion can be added to a user's image based at least in part on a combination of the user's prescription and the glasses frames shape. Even more particularly, facial distortion can be added outward or inward based on the appropriate form of distortion. For example, outward facial distortion can include when the glasses expands a user's face into the glasses lens, thus distorting the appearance of the user. As another example, inward facial distortion can include when the glasses shrinks a user's face into the glasses lens, thus distorting the appearance of the user.

In some implementations, the computing system can further alter the user image. For example, the computing system can remove glasses frames from the user image. As yet another example, the computing system can add shading to the glasses frames or glasses lenses in a user image. In particular, the shading to the glasses frames or glasses lenses can be a particular color (e.g., user determined). Even more particularly, the user can determine how much shading (opaque vs translucent) is applied (e.g., in a sliding scale gradient).

In some implementations, the computing system can generate a modified user image such that a plurality of modified user images can be generated. In particular, the plurality of modified user images can each be associated with a different camera angle (e.g., a different camera angle of the user in the user's modified state). Even more particularly, the different camera angles can be associated with associated different camera angles of glasses frames present in the modified user images. For example, each of the plurality of modified images can be associated with a corresponding different camera angle of the glasses frames on a user in different camera angles. Specifically, the plurality of modified images can include associating a corresponding modified facial displacement distortion. Even more specifically, the corresponding modified facial displacement distortion can be determined based on the associated camera angle.

In some implementations, the computing system can continuously update a video in real-time (e.g., by removing glasses frames, adding glasses frames, adding facial distortion, removing facial distortion). For example, the computing system can update a user feed in a video conference such that a user wearing glasses can have facial distortion removed during the video conference. As another example, the computing system can update a user feed in a video conference such that a user not wearing glasses can appear to be wearing glasses (e.g., the computing system can add facial distortion) during the video conference. As yet another example, the computing system can update a user feed in a virtual shopping application such that a user can view a live feed of themselves trying on glasses frames (e.g., with or without facial distortion).

In some implementations, photo compilations generated by a user can have images corrected (e.g., automatically). For example, a user can indicate to the computing system that the user desires to correct facial distortion present in all or a selection of photos within a compilation (e.g., collage, album, etc.) Specifically, the computing system can automatically detect whether an image contains a person wearing glasses and subsequently automatically generate a modified image where the person wearing glasses does not have facial distortion due to wearing glasses.

In some implementations, the computing system can request an input of a user's background image (e.g., if the facial displacement is outward). As another example, the computing system can request an input of an image of the user without glasses frames. In particular, the computing system can request an input of a series of images from several frontal angles of the user without glasses. As yet another example, the computing system can request an input of an image of the user with personal glasses frames with lenses. In particular, the computing system can request an input of a series of images from several angles of the user wearing personal glasses frames with lenses.

In some implementations, the computing system can detect the distortion area. In some implementations, the computing system can detect the angle of the user's facial position. In particular, the computing system can determine the distortion area and angle of the user's facial position simultaneously or consecutively.

With reference now to the Figures, example embodiments of the present disclosure will be discussed in further detail.

Example Devices and Systems

Figure 1 depicts a block diagram of an example computing system 100 according to example embodiments of the present disclosure. The system 100 includes a user computing device 102, a server computing system 130, and an image transformation computing system 150 that are communicatively coupled over a network 180.

The user computing device 102 can be any type of computing device, such as, for example, a personal computing device (e.g., laptop or desktop), a mobile computing device (e.g., smartphone or tablet), a gaming console or controller, a wearable computing device, an embedded computing device, or any other type of computing device.

The user computing device 102 includes one or more processors 112 and a memory 114. The one or more processors 112 can be any suitable processing device (e.g., a processor core, a microprocessor, an ASIC, a FPGA, a controller, a microcontroller, etc.) and can be one processor or a plurality of processors that are operatively connected. The memory 114 can include one or more non-transitory computer-readable storage mediums, such as RAM, ROM, EEPROM, EPROM, flash memory devices, magnetic disks, etc., and combinations thereof. The memory 114 can store data 116 and instructions 118 which are executed by the processor 112 to cause the user computing device 102 to perform operations.

In some implementations, the user computing device 102 can store or include one or more machine-learned model(s) 120. For example, the machine-learned model(s) 120 can be or can otherwise include various machine-learned models such feed-forward neural networks, recurrent neural networks (e.g., long short-term memory recurrent neural networks), convolutional neural networks, residual neural networks, or other forms of neural networks.

In some implementations, the machine-learned model(s) 120 can be received from the server computing system 130 over network 180, stored in the user computing device memory 114, and then used or otherwise implemented by the one or more processors 112. In some implementations, the user computing device 102 can implement multiple parallel instances of a single neural network 120

Additionally or alternatively, machine-learned model(s) 140 can be included in or otherwise stored and implemented by the server computing system 130 that communicates with the user computing device 102 according to a client-server relationship. For example, the machine-learned model(s) 140 can be implemented by the server computing system 140 as a portion of a web service. Thus, machine-learned model(s) 120 can be stored and implemented at the user computing device 102 and/or one or more networks 140 can be stored and implemented at the server computing system 130.

The user computing device 102 can also include one or more user input component 122 that receives user input. For example, the user input component 122 can be a touch-sensitive component (e.g., a touch-sensitive display screen or a touch pad) that is sensitive to the touch of a user input object (e.g., a finger or a stylus). The touch-sensitive component can serve to implement a virtual keyboard. Other example user input components include a microphone, a traditional keyboard, or other means by which a user can provide user input. The user input can be used, in some implementations, to specify one or more desired values for a user-specified conditioning vector, which will be discussed in greater detail with reference to FIGS. 3 and 4.

The server computing system 130 includes one or more processors 132 and a memory 134. The one or more processors 132 can be any suitable processing device (e.g., a processor core, a microprocessor, an ASIC, a FPGA, a controller, a microcontroller, etc.) and can be one processor or a plurality of processors that are operatively connected. The memory 134 can include one or more non-transitory computer-readable storage mediums, such as RAM, ROM, EEPROM, EPROM, flash memory devices, magnetic disks, etc., and combinations thereof. The memory 134 can store data 136 and instructions 138 which are executed by the processor 132 to cause the server computing system 130 to perform operations.

In some implementations, the server computing system 130 includes or is otherwise implemented by one or more server computing devices. In instances in which the server computing system 130 includes plural server computing devices, such server computing devices can operate according to sequential computing architectures, parallel computing architectures, or some combination thereof.

As described above, the server computing system 130 can store or otherwise include machine-learned model(s) 140. For example, the machine-learned model(s) 140 can be or can otherwise include feed forward neural networks, deep neural networks, recurrent neural networks, residual neural networks, and convolutional neural networks.

The user computing device 102 and/or the server computing system 130 can train and/or evaluate the machine-learned model(s) 120 and/or 140 via interaction with the image transformation computing system 150 that is communicatively coupled over the network 180. The image transformation computing system 150 can be separate from the server computing system 130 or can be a portion of the server computing system 130.

The image transformation computing system 150 includes one or more processors 152 and a memory 154. The one or more processors 152 can be any suitable processing device (e.g., a processor core, a microprocessor, an ASIC, a FPGA, a controller, a microcontroller, etc.) and can be one processor or a plurality of processors that are operatively connected. The memory 154 can include one or more non-transitory computer-readable storage mediums, such as RAM, ROM, EEPROM, EPROM, flash memory devices, magnetic disks, etc., and combinations thereof. The memory 154 can store data 156 and instructions 158 which are executed by the processor 152 to cause the image transformation computing system 150 to perform operations. In some implementations, the image transformation computing system 150 includes or is otherwise implemented by one or more server computing devices.

The image transformation computing system 150 can include a model trainer 160 that trains and/or evaluates the machine-learned model(s) 120 and/or 140 stored at the user computing device 102 and/or the server computing system 130 using various training or learning techniques, such as, for example, backwards propagation of errors. In some implementations, performing backwards propagation of errors can include performing truncated backpropagation through time. The model trainer 160 can perform a number of generalization techniques (e.g., weight decays, dropouts, etc.) to improve the generalization capability of the models being trained.

In particular, the model trainer 160 can train the machine-learned model(s) 120 and/or 140 based on a set of training data 162. The training data 162 can be, but is not limited to, unpaired training data (e.g., sets of images sharing one or more defined characteristics, such as depicting a winter environment at night).

The image transformation computing system 150 can also include image transforming model(s) 159. In some implementations, the image transforming model(s) 159 can include machine-learned generator model(s) and machine-learned discriminator model(s) configured to perform image transformations on an image based on a user-specified conditioning vector. In some implementations, the machine-learned generator model(s) and the machine-learned discriminator model(s) of the image transforming model(s) 159 can be trained by the model trainer 160 in an adversarial fashion (e.g., as a generative adversarial network (GAN), etc.). The image transformation computing system 150 can also optionally be communicatively coupled with various other devices (e.g., the user computing device 102) to provide trained image transformation model(s) (e.g., machine-learned generator models, machine-learned discriminator model(s), etc.) to the various other devices and/or to receive data from various other devices (e.g., receiving an image from the user computing device 102 as an input to the image transformation model(s) 159, sending the transformed image to the computing device 102, etc.).

Each of the model trainer 160 and the image transformation model(s) 159 can include computer logic utilized to provide desired functionality. Each of the model trainer 160 and the image transformation model(s) 159 can be implemented in hardware, firmware, and/or software controlling a general purpose processor. For example, in some implementations, each of the model trainer 160 and the image transformation model(s) 159 can include program files stored on a storage device, loaded into a memory and executed by one or more processors. In other implementations, each of the model trainer 160 and the network searcher 159 can include one or more sets of computer-executable instructions that are stored in a tangible computer-readable storage medium such as RAM hard disk or optical or magnetic media.

The network 180 can be any type of communications network, such as a local area network (e.g., intranet), wide area network (e.g., Internet), or some combination thereof and can include any number of wired or wireless links. In general, communication over the network 180 can be carried via any type of wired and/or wireless connection, using a wide variety of communication protocols (e.g., TCP/IP, HTTP, SMTP, FTP), encodings or formats (e.g., HTML, XML, JSON, YAML), and/or protection schemes (e.g., VPN, secure HTTP, SSL).

Figure 1 illustrates one example computing system that can be used to implement the present disclosure. Other computing systems can be used as well. For example, in some implementations, the user computing device 102 can include the model trainer 160 and the training dataset 162. In such implementations, the networks 120 can be both trained and used locally at the user computing device 102. In some of such implementations, the user computing device 102 can implement the model trainer 160 to personalize the networks 120 based on user-specific data.

Further, although the present disclosure is described with particular reference to neural networks. The systems and methods described herein can be applied to other multi-layer machine-learned model architectures.

Figure 2 depicts a block diagram of an example computing device 10 according to example embodiments of the present disclosure. The computing device 10 can be a user computing device or a server computing device.

The computing device 10 includes a number of applications (e.g., applications 1 through N). Each application contains its own machine learning library and machine-learned model(s). For example, each application can include machine-learned generator model(s) and/or machine-learned discriminator model(s). Example applications include a text messaging application, an email application, a dictation application, a virtual keyboard application, a browser application, an image capture application, an image transformation application, an image upload application, etc.

As illustrated in Figure 2, each application can communicate with a number of other components of the computing device, such as, for example, one or more sensors, a context manager, a device state component, and/or additional components. In some implementations, each application can communicate with each device component using an API (e.g., a public API). In some implementations, the API used by each application is specific to that application.

Figure 3 depicts a block diagram of an example computing device 50 according to example embodiments of the present disclosure. The computing device 50 can be a user computing device or a server computing device.

The computing device 50 includes a number of applications (e.g., applications 1 through N). Each application is in communication with a central intelligence layer. Example applications include a text messaging application, an email application, a dictation application, a virtual keyboard application, a browser application, an image capture application, an image transformation application, an image upload application, etc. In some implementations, each application can communicate with the central intelligence layer (and model(s) stored therein) using an API (e.g., a common API across all applications).

The central intelligence layer includes a number of machine-learned models. For example, as illustrated in Figure 3, a respective machine-learned model (e.g., a machine-learned generator model, a machine-learned discriminator model, etc.) can be provided for each application and managed by the central intelligence layer. In other implementations, two or more applications can share a single machine-learned model. For example, in some implementations, the central intelligence layer can provide a single model (e.g., a single model) for all of the applications. In some implementations, the central intelligence layer is included within or otherwise implemented by an operating system of the computing device 50.

The central intelligence layer can communicate with a central device data layer. The central device data layer can be a centralized repository of data for the computing device 50. As illustrated in Figure 3, the central device data layer can communicate with a number of other components of the computing device, such as, for example, one or more sensors, a context manager, a device state component, and/or additional components. In some implementations, the central device data layer can communicate with each device component using an API (e.g., a private API).

Example Training Environment and Methods

Figure 4 depicts a graphical diagram of example machine-learned models for output image generation and associated training signals according to example embodiments of the present disclosure. An input image 406 and user-specified conditioning vector 404 can be received as an input by machine-learned generator model(s) 408. The input image 406 can be a digital image (e.g., a digital image captured by a digital image capture device, a scanned image, etc.). In some implementations, the input image 406 can be an output of machine-learned generator model(s), as will be discussed in greater detail with regards to figure 5. The user specified conditioning vector 404 can parameterize one or more desired values for one or more defined characteristics of an output image. In some implementations, the user-specified conditioning vector 404 can be a real and continuously valued vector. As an example, the vector can include values (e.g., parameterized user-desired values) that are both real and continuous (e.g., 0.5, 1, .2235, -5, etc.), the values corresponding to one or more desired characteristics of the output image.

In some implementations, the user-specified conditioning vector 404 can describe a degree to which the one or more transformations are applied to the input image. As an example, the user-specified conditioning vector 404 can include parameterized user-desired values (e.g., 1, 0, 0.5, etc.) that describe one or more transformations to perform and the desired degree to which the transformation(s) (e.g., transforming within glasses frames to include facial displacement distortion, etc.) should be performed. For example, the conditioning vector 404 can describe a facial displacement distortion transformation that is to be applied at a value of 0.5, resulting in a partially transformed output image. As another example, the conditioning vector 404 can describe a facial displacement distortion transformation to be applied at a value of 1.0, resulting in a fully transformed output image.

In some implementations, the user-specified conditioning vector 404 can specify one or more areas of the input image 406 to which the one or more transformations are applied. As an example, the user-specified conditioning vector 404 can include value(s) that specify that the transformation(s) should be applied to a left half of the input image 406. As another example, the user-specified conditioning vector 404 can include value(s) that specify that the transformation should be applied to a left half portion of the input image 406.

The machine-learned generator model(s) 408 can include an encoder portion 408A and a decoder portion 408B. The user-specified conditioning vector 404 can be provided to the decoder portion 408B of the one or more machine-learned generator models 408. More particularly, a series of layers in the encoder portion 408A (e.g., convolutional layers in a convolutional neural network) can encode the input image 406 into a compressed representation that can be used as inputs to a series of residual blocks (e.g., residual blocks of a residual neural network(s)). A stack of connected layers can learn to transform the raw values of the user-specified conditioning vector 404 to an alternative representation. At the lowest layer of the machine-learned generator model(s) 408, the image representation (e.g., the lower-dimensional representation of the input image 406) and the alternative representation of the user-specified conditioning vector 404 can be mixed by concatenation to one or more areas along the feature dimension of the input image 406. A series of convolutional layers can decode the combined (e.g., concatenated) data into an output image 412.

The one or more machine-learned generator models 408 can be or otherwise include one or more neural networks (e.g., deep neural networks) or the like. Neural networks (e.g., deep neural networks) can be feed-forward neural networks, convolutional neural networks, residual neural networks, and/or various other types of neural networks. In some implementations, the one or more machine-learned generator models 408 can be residual neural networks including connections (e.g., skip connections) between individual layers of the residual neural networks. The connections between layers can be utilized to send the user-specified conditioning vector to a matching layer (e.g., a mirrored layer) of the network before the user-specified conditioning vector is reduced to a lowest alternative representation. In such fashion, a training signal (e.g., a discriminator output 414) can be more effectively backpropagated through the machine-learned generator model(s).

The machine-learned generator model(s) 408 can be configured to perform, based at least in part on the user-specified conditioning vector 404, one or more transformations on the input image 406 to generate an output image 412 with the one or more desired values for one or more defined characteristics of the output image 412. The user-specified conditioning vector 404 can specify the transformation(s) to be performed on the input image 406. As an example, the user-specified conditioning vector 404 can specify a facial displacement distortion transformation (e.g., transforming within the glasses frames to include facial displacement distortion, removing the effect of the glasses distortion) to apply to the input image 406. As another example, the user-specified conditioning vector 404 can specify a lack of glasses frames-to-glasses frames transformation and a facial displacement distortion transformation (e.g., transforming the user’s face depicted in the input image to add in glasses frames and then add the facial displacement distortion transformation). The user-specified conditioning vector 404 can describe a plurality of transformations and a plurality of defined characteristics associated with each respective transformation.

In some implementations, the defined characteristics can include light characteristics and/or light source location characteristics. As an example, a digital input image 406 depicting a human face can be transformed so that a light source is present in the image projecting light at a certain intensity. As will be discussed in greater detail with reference to Figure 6B, the light source can be placed three-dimensionally during the transformation. For example, the desired values for the light source may place the light source behind the human face at a relative intensity 0.2. For another example, the desired values for the light source may place the light source in front and to the right of the human face at a relative intensity of 0.8. In such fashion, the light source transformation can illuminate and/or darken various aspects of the input image 406 in a three-dimensional manner. It should be noted that a plurality of light sources can be included in the light source transformation based on the desired values for the defined light source transformation characteristics.

In some implementations, the defined characteristics can include color characteristics. As an example, an input image 406 with colored glasses frames (e.g., black and white, grayscale, etc.) can be transformed to an output image 412 containing glasses frames colored in an alternate fashion (e.g., solid colors, prints, patterns, etc.). In some implementations, the defined characteristics can include user prescription characteristics. Prescription characteristics can include any sort of glasses prescription. As an example, a stronger prescription can indicate that a larger facial displacement distortion is applied in output image 412.

The machine-learned discriminator model 410 can be configured to receive the output image 412, a target image from target image set 402, and the user-specified conditioning vector 404. A target image 402 can be an image from a set of target images representative of the desired output image 412. As an example, if the output image 412 has been transformed to include defined characteristics of a particular prescription, the target image 402 can be an image from a set of users wearing glasses of the particular prescription.

The machine-learned discriminator model(s) 410 can be configured to generate a discriminator output 414 that selects one of the output image 412 and the target image 402 as having the one or more desired values for the one or more defined characteristics. The machine-learned discriminator model(s) 410 can account for the inclusion of the user-specified conditioning vector 404 in the transformation. More particularly, the machine-learned discriminator model(s) 410 can, based at least in part on the user-specified conditioning vector 404, select one of the output image 412 and the target image 402 based on the defined characteristics of the output image 412.

The one or more machine-learned discriminator models 410 can contain a plurality of layers configured to evaluate different aspects of the discriminator output 414. As an example, an initial layer (e.g., a convolutional layer of a convolutional neural network) of the machine-learned discriminator model(s) 410 can operate solely on the output image 412 to extract a suitable deep representation (e.g., a latent space representation of the output image 412). The machine-learned discriminator model(s) 410 can transform the user-specified conditioning vector 404 through a series of fully connected layers (e.g., convolutional layers of a convolutional neural network) in a similar fashion to the machine-learned generator model(s) 408. The learned representations of the input image 406 and of the user-specified conditioning vector 404 can be mixed by concatenation to spatial location(s) along of the feature dimension in a manner similar to the machine-learned generator model(s) 408. A series of layers (e.g., convolutional layers of a convolutional neural network) can operate on the concatenated representation to produce a classification score for the patch(s) of the output image 412. The selection output (e.g., the discriminator output 414) can be obtained by analyzing patch(s) of the output image 412 (e.g., performing a mean average pooling operation(s), etc.).

The one or more machine-learned discriminator models 410 can be or otherwise include one or more neural networks (e.g., deep neural networks) or the like. Neural networks (e.g., deep neural networks) can be feed-forward neural networks, convolutional neural networks, residual neural networks, and/or various other types of neural networks. In some implementations, the machine-learned discriminator model(s) 410 and machine-learned generator model(s) 408 can be components of a generative adversarial network (GAN). In some implementations, the one or more machine-learned discriminator models 410 can be residual neural networks including connections (e.g., skip connections) between individual layers of the residual neural networks. The user-specified conditioning vector 404 can be sent through the connections between layers from a first layer to a matching layer (e.g., a mirrored layer) of the network before the user-specified conditioning vector 404 is reduced to a lowest alternative representation. In such fashion, a training signal (e.g., the discriminator output 414) can be evaluated with an objective function and more effectively backpropagated through the machine-learned discriminator model(s) 410. As an example, the machine-learned discriminator model(s) 410 can be trained on a difference between a ground truth associated with the discriminator output 414 (e.g., a defined transformation quality of the output image 412) and the discriminator output 414 from the machine-learned discriminator model(s) 410.

The one or more machine-learned generator models 408 can be trained based at least in part on an objective function evaluation of the discriminator output 414 using various training or learning techniques, such as, for example, backwards propagation of errors (e.g., truncated backpropagation through time). In some implementations, the machine-learned discriminator model(s) 410 and machine-learned generator model(s) 414 can be trained in an adversarial fashion (e.g., a generative adversarial network). As one example, in some implementations, training the machine-learned generator model(s) based on the discriminator output 414 can include performing stochastic gradient descent to optimize an objective function that evaluates the discriminator output 414 (e.g., minimize the discriminator output 414, etc.). In such fashion, the machine-learned discriminator model(s) 410 can be trained to optimize the value of the discriminator output 414 while the machine-learned generator model(s) can be trained to minimize the value of the discriminator output 414 (e.g., in an adversarial fashion).

Figure 5 depicts a graphical diagram of example machine-learned models for reconstructive input image generation and associated training signals according to example embodiments of the present disclosure. In some implementations, as depicted, machine-learned generator model(s) and machine-learned discriminator model(s) can be run in a mirrored configuration with respect to the model architecture depicted in Figure 4. More particularly, machine-learned generator model(s) 508 and machine-learned discriminator model(s) 510 can be run in a parallel configuration to perform transformations on the output image 412 (e.g., the output image 412 generated by machine-learned generator model(s) 408).

The machine-learned generator model(s) 508 (e.g., including encoder portion 508A and decoder portion 508B) can receive output image 412 and user-specified conditioning vector 414 as inputs. The machine-learned generator model(s) 508 can produce a reconstructed output image 512 in the same fashion as machine-learned generator model(s) 408 of Figure 4. The reconstructed input image 512 can be a reconstruction of the input image 406 from the output image 412. More specifically, by generating the reconstructed input image 512 based on the output image 412 and the user-specified conditioning vector 404, the machine-learned generator model(s) 508 can reverse the parameterized transformations specified by the vector and initially applied by machine-learned generator model(s) 408.

As an example, input image 406 can be an image depicting a user wearing glasses frames with planar lenses which may not distort the user image. The machine-learned generator model(s) 408 of Figure 4 can, based on the user specified conditioning vector 414 and the input image 406, generate an output image 412 depicting a user wearing glasses frames with facial displacement distortion. The machine-learned generator model(s) 508 can receive the output image 412 and the user-specified conditioning vector 404 and generate, as an output, a reconstructed input image 512. The reconstructed input image 512 can depict a reconstruction of the depiction of the input image 406 (e.g., the user wearing glasses frames with planar lenses). In such fashion, the machine-learned generator model(s) 508 can reconstruct the input image from the output image by applying a transformation that reverses the transformation applied by machine-learned generator model(s) 408.

The reconstructed input image 512 and the input image 406 can be received by the machine-learned discriminator model(s) 510 as inputs. The machine-learned discriminator model(s) 510 can evaluate a difference between the reconstructed input image 512 and the input image 406 to output a reconstructive discriminator output 514. The reconstructive discriminator output can be used as a training signal for an optimization function. The optimization function can evaluate the reconstructive discriminator output 414 and, based at least in part on the reconstructive discriminator output 414, modify values for one or more parameters of the machine-learned discriminator model(s) 510 and/or the machine-learned generator model(s) 508 based on the optimization function. More particularly, the output can be backpropagated through the machine-learned model(s) (e.g., 508 and 510) to determine values associated with one or more parameters of the model(s) to be updated. The one or more parameters can be updated to reduce the difference evaluated by the optimization function (e.g., using an optimization procedure, such as a gradient descent algorithm).

In some implementations, the reconstructive discriminator output 514 can be used as a training signal to the machine-learned generator model(s) 408 and machine-learned discriminator model(s) 410 of Figure 4. Based at least in part on an optimization function that evaluates the reconstructive discriminator output 514, values for one or more parameters of the machine-learned discriminator model(s) 410 and/or the machine-learned generator model(s) can be modified 408. More particularly, the output can be backpropagated through the machine-learned model(s) to determine values associated with one or more parameters of the model(s) to be updated. The one or more parameters can be updated to reduce the difference evaluated by the optimization function (e.g., using an optimization procedure, such as a gradient descent algorithm).

In such fashion, the performance of the mirrored model architecture (e.g., model(s) 508 and 510) can be used as a training signal for the transformational model architecture (e.g., model(s) 408 and 410) to enforce consistency between transformations. More particularly, the addition of the mirrored model architecture can serve to enforce structure on the generated outputs of machine-learned generator model 408 to minimize deviation of the transformations.

Figure 6 depicts a graphical representation of an example facial distortion image transformation according to example embodiments of the present disclosure. 600 depicts an example transformation performed by the machine-learned generator model(s) of the present disclosure. As depicted, a machine-learned generator model(s) can receive a user-specified conditioning vector. The user-specified conditioning vector can include parameterized values the user desires for defined characteristics of the transformation(s) to be applied. As a specific example, a 100% applied facial distortion due to a particular glasses prescription 602 can be seen. Specifically the applied facial distortion due to a particular glasses prescription 602 is applied inward. However, although not shown, the applied facial distortion due to a particular glasses prescription 602 can be applied outward as well. As another example, the machine-learned generator model can apply glasses frames 604 to an image of a user 606.

Additional Disclosure

The technology discussed herein makes reference to servers, databases, software applications, and other computer-based systems, as well as actions taken and information sent to and from such systems. The inherent flexibility of computer-based systems allows for a great variety of possible configurations, combinations, and divisions of tasks and functionality between and among components. For instance, processes discussed herein can be implemented using a single device or component or multiple devices or components working in combination. Databases and applications can be implemented on a single system or distributed across multiple systems. Distributed components can operate sequentially or in parallel.

While the present subject matter has been described in detail with respect to various specific example embodiments thereof, each example is provided by way of explanation, not limitation of the disclosure. Those skilled in the art, upon attaining an understanding of the foregoing, can readily produce alterations to, variations of, and equivalents to such embodiments. Accordingly, the subject disclosure does not preclude inclusion of such modifications, variations and/or additions to the present subject matter as would be readily apparent to one of ordinary skill in the art. For instance, features illustrated or described as part of one embodiment can be used with another embodiment to yield a still further embodiment. Thus, it is intended that the present disclosure cover such alterations, variations, and equivalents.

Abstract

A computing system and method that can be used for performing transformations of images using machine-learned model(s) and interpretable user-specified parameters that control the transformations. More specifically, the machine-learned model can be used to perform transformation(s) on an image based on different parameter(s) of a user-specified conditioning vector. The systems and methods of the present disclosure allow for a user to specify desired values for defined characteristics of a transformation (e.g., area of transformation, degree of transformation, etc.) for one or more transformations of the defined characteristic(s) (e.g., transforming lack of facial displacement distortion to presence of a facial displacement distortion, transforming glasses frames to lack of glasses frames, unshaded to shaded, etc.).

Figures

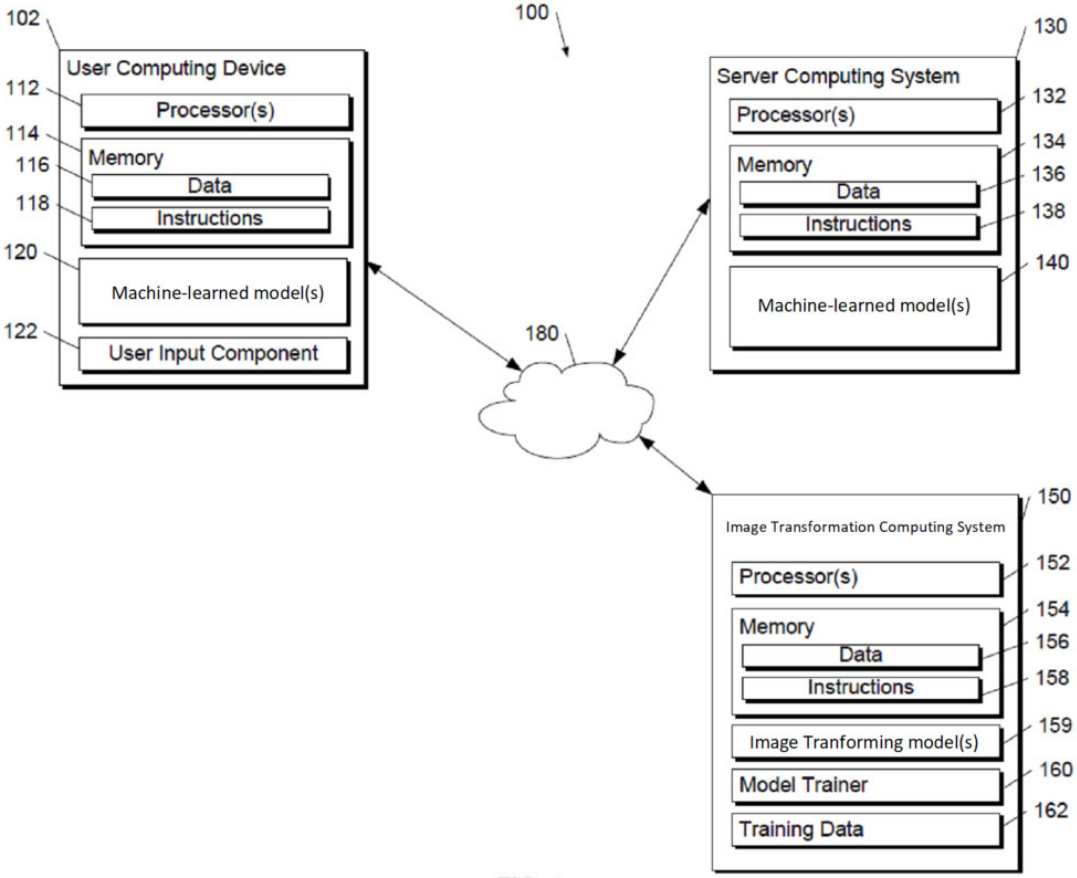

FIG. 1

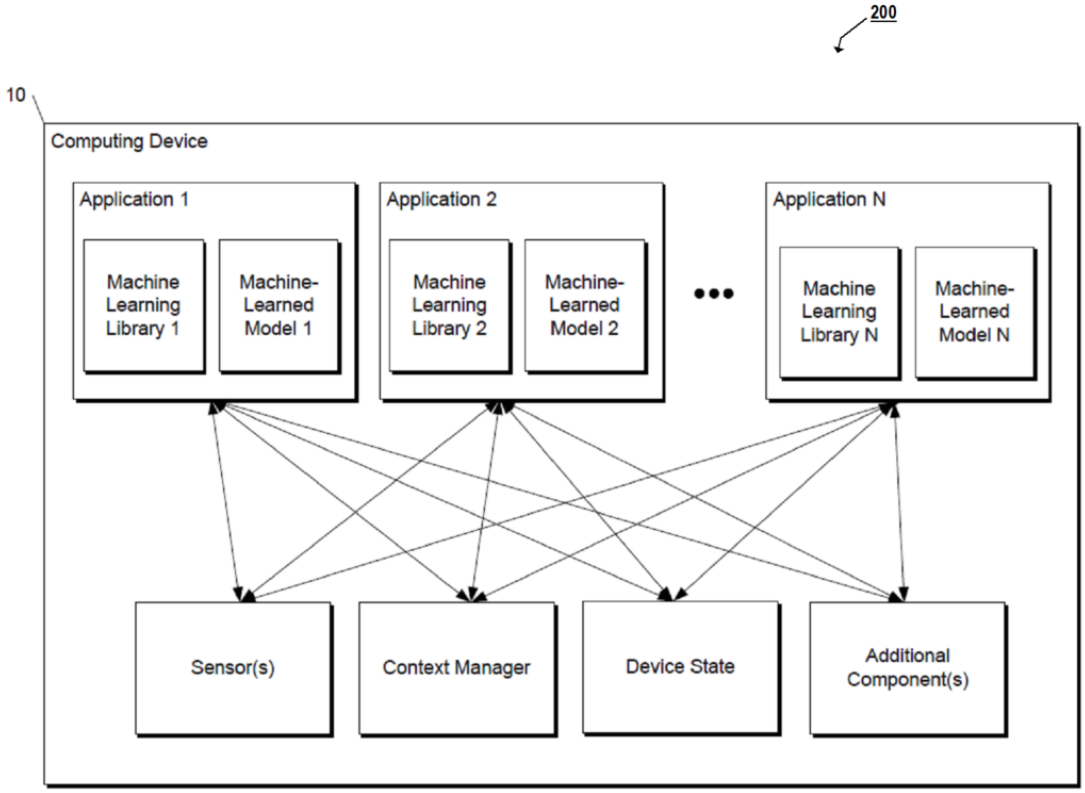
200
10
Computing Device
Application 1
Machine Learning Library 1
Machine-Learned Model 1
Application 2
Machine Learning Library 2
Machine-Learned Model 2
Application N
Machine Learning Library N
Machine-Learned Model N
Sensor(s)
Context Manager
Device State
Additional Component(s)

FIG. 2

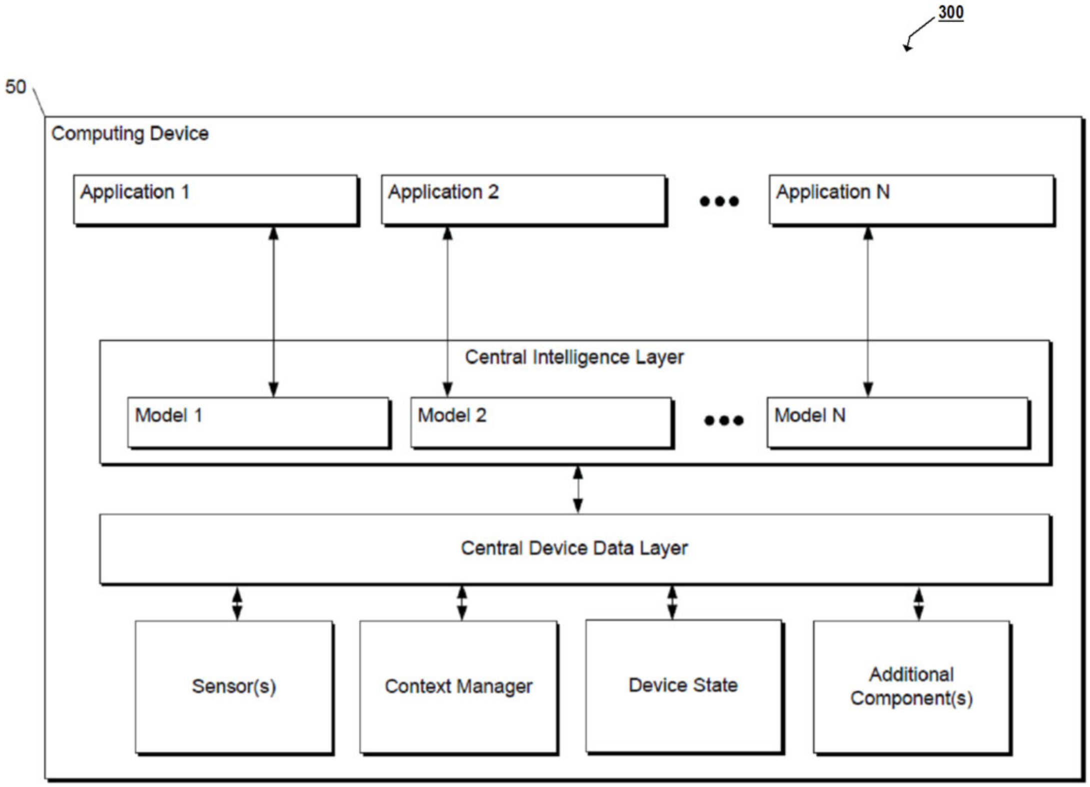
300
50
Computing Device
Application 1
Application 2
•••
Application N
Central Intelligence Layer
Model 1
Model 2
•••
Model N
Central Device Data Layer
Sensor(s)
Context Manager
Device State
Additional Component(s)

FIG. 3

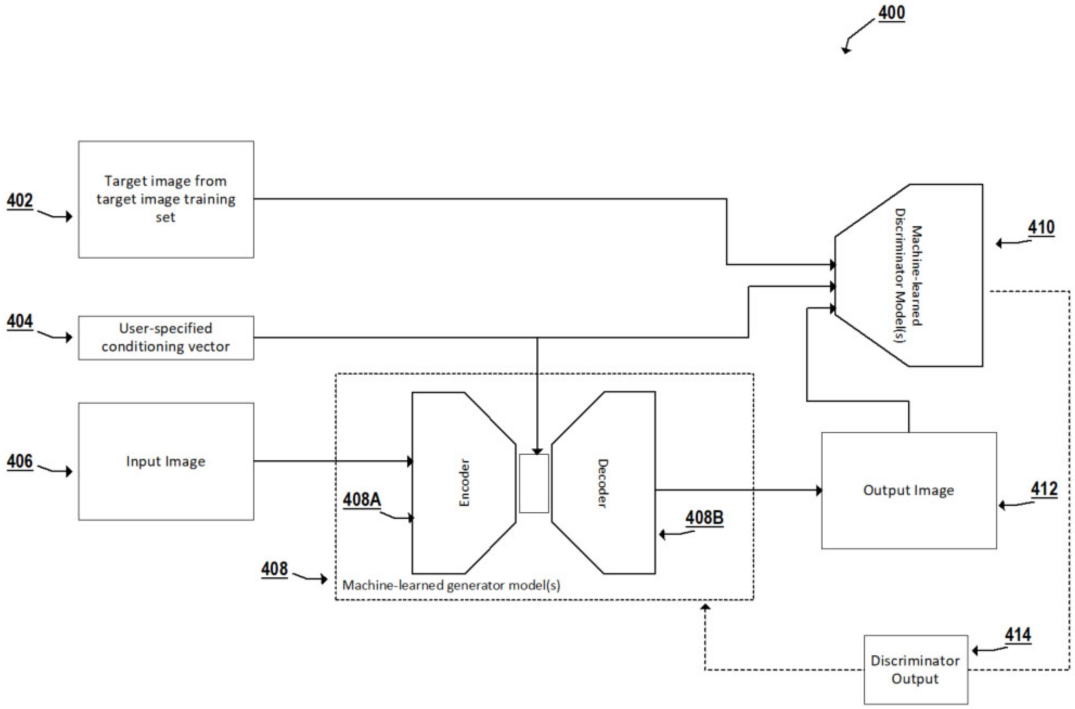
400
402
Target image from target image training set
404
User-specified conditioning vector
406
Input Image
408A
Encoder
Decoder
408B
408
Machine-learned generator model(s)
Machine-learned Discriminator Model(s)
410
Output Image
412
Discriminator Output
414

FIG. 4

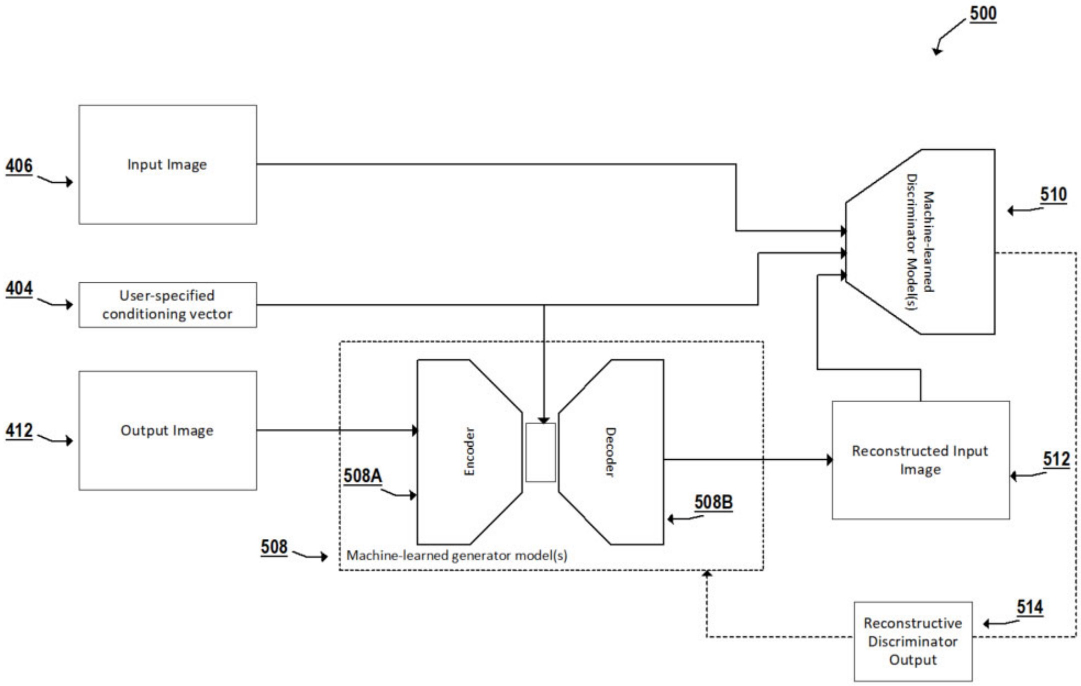
500
406
Input Image
404
User-specified conditioning vector
412
Output Image
508A
Encoder
Decoder
508B
508
Machine-learned generator model(s)
Machine-learned Discriminator Model(s)
510
Reconstructed Input Image
512
514
Reconstructive Discriminator Output

FIG. 5

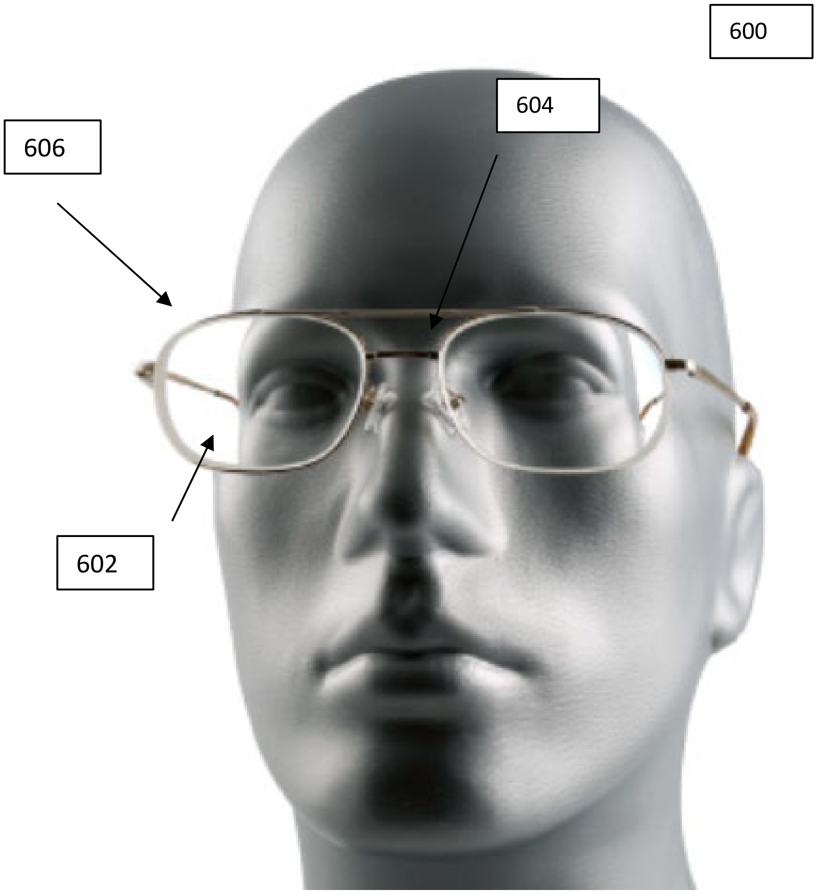

FIG. 6