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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 May 2024 (16.05.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/102433 A1

(51) International Patent Classification:
G06V 40/20 (2022.01) *G16H 20/30* (2018.01)
G06V 10/82 (2022.01) *G06N 3/096* (2023.01)
G06V 10/764 (2022.01)

(21) International Application Number:
PCT/US2023/037076

(22) International Filing Date:
09 November 2023 (09.11.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/424,155 10 November 2022 (10.11.2022) US
63/425,327 15 November 2022 (15.11.2022) US
18/504,469 08 November 2023 (08.11.2023) US

(71) Applicant: **NEC LABORATORIES AMERICA, INC.** [US/US]; 4 Independence Way, Suite 200, Princeton, New Jersey 08540 (US).

(72) Inventors: **LI, Kai**; 515 Raven Crest Drive, Plainsboro, New Jersey 08536 (US). **PATEL, Deep**; 20 Delar Parkway, Franklin Park, New Jersey 08823 (US). **KRUUS, Erik**; 261 Hamilton Road East, Hillsborough, New Jersey 08844 (US). **MIN, Renqiang**; 7 Carlton Circle, Princeton, New Jersey 08540 (US).

(74) Agent: **BITETTO, James J.**; 401 Broadhollow Road, Suite 402, Melville, New York 11747 (US).

(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,

WO 2024/102433 A1

(54) Title: MACHINE LEARNING OF SPATIO-TEMPORAL MANIFOLDS FOR SOURCE-FREE VIDEO DOMAIN ADAPTATION

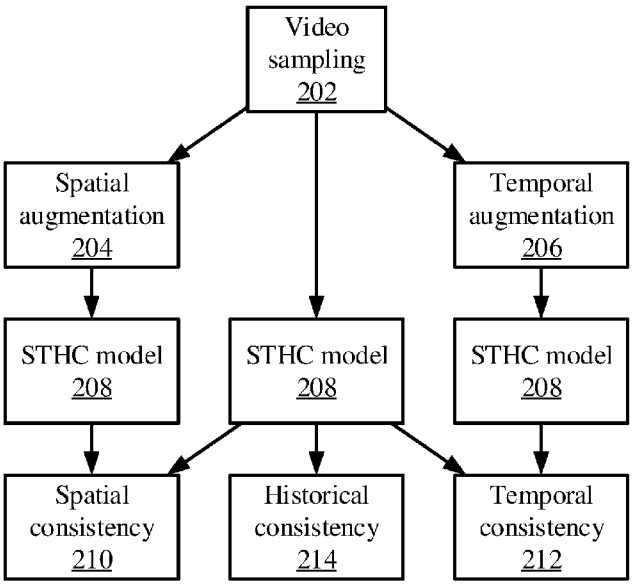


FIG. 2

(57) Abstract: Methods and systems for training a model include performing spatial augmentation (204) on an unlabeled input video to generate spatially augmented video. Temporal augmentation is performed (206) on the input video to generate temporally augmented video. Predictions are generated (208), using a model that was pre-trained on a labeled dataset, for the unlabeled input video, the spatially augmented video, and the temporally augmented video. Parameters of the model are adapted (404) using the predictions while enforcing spatial consistency, temporal consistency, and historical consistency. The model may be used for action recognition in a healthcare context, with recognition results being used for determining whether patients are performing a rehabilitation exercise correctly.

[Continued on next page]

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).

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

MACHINE LEARNING OF SPATIO-TEMPORAL MANIFOLDS FOR SOURCE-FREE VIDEO DOMAIN ADAPTATION

RELATED APPLICATION INFORMATION

[0001] This application claims priority to U.S. Patent Application No. 63/424,155, filed on November 10, 2022, and to U.S. Patent Application No. 63/425,327, filed on November 15, 2022, and U.S. Patent Application No. 18/504,469 filed on November 8, 2023, each incorporated herein by reference in its entirety.

BACKGROUND

Technical Field

[0002] The present invention relates to action recognition and, more particularly, to learning manifolds for video domain adaptation.

Description of the Related Art

[0003] Action recognition is a task used in automated video understanding that identifies an action that is being performed in a video sequence. However, an action recognition model that is trained with annotated data drawn from one statistical distribution may show decreased performance when tested on data outside that distribution.

SUMMARY

[0004] A method for training a model includes training a model include performing spatial augmentation on an unlabeled input video to generate spatially augmented video. Temporal augmentation is performed on the input video to generate temporally augmented video. Predictions are generated, using a model that was pre-trained on a labeled dataset, for the unlabeled input video, the spatially augmented video, and the

temporally augmented video. Parameters of the model are adapted using the predictions while enforcing temporal consistency, temporal consistency, and historical consistency.

[0005] A system for training a model includes a hardware processor and a memory that stores a computer program. When executed by the hardware processor, the computer program causes the hardware processor to perform spatial augmentation on an unlabeled input video to generate spatially augmented video, to perform temporal augmentation on the input video to generate temporally augmented video, to generate predictions, using a model that was pre-trained on a labeled dataset, for the unlabeled input video, the spatially augmented video, and the temporally augmented video, and to adapt parameters of the model using the predictions while enforcing temporal consistency, temporal consistency, and historical consistency.

[0006] These and other features and advantages will become apparent from the following detailed description of illustrative embodiments thereof, which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF DRAWINGS

[0007] The disclosure will provide details in the following description of preferred embodiments with reference to the following figures wherein:

[0008] FIG. 1 is a diagram of a video that includes a series of frames portraying an action, in accordance with an embodiment of the present invention;

[0009] FIG. 2 is a diagram of a training framework for an action recognition model that includes spatial, temporal, and historical consistency constraints, in accordance with an embodiment of the present invention;

[0010] FIG. 3 is pseudo-code for updating the parameters of a pre-trained action recognition model based on spatial, temporal, and historical consistency, in accordance with an embodiment of the present invention;

[0011] FIG. 4 is a block/flow diagram of a method for training and using an action recognition model, in accordance with an embodiment of the present invention;

[0012] FIG. 5 is a block diagram of an exemplary application of an action recognition model in a healthcare facility, in accordance with an embodiment of the present invention;

[0013] FIG. 6 is a block/flow diagram illustrating stages in an action recognition model, in accordance with an embodiment of the present invention;

[0014] FIG. 7 is a block diagram of a computing device that can train and use an action recognition model, in accordance with an embodiment of the present invention;

[0015] FIG. 8 is a diagram of a neural network architecture that can be used to implement part of the action recognition model, in accordance with an embodiment of the present invention; and

[0016] FIG. 9 is a diagram of a deep neural network architecture that can be used to implement part of the action recognition model, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0017] Unsupervised video domain adaptation (UVDA) learns an adaptive model using labeled video data from a source domain and unlabeled video data from a target domain. UVDA may take input videos from both domains and train a model by minimizing a classification risk on labeled source videos, aligning videos from both domains in a class-agnostic fashion. As described herein, spatio-temporal manifolds

(STEMs) may be learned, where actions of a same category may be drawn from the same low-dimensional manifold, regardless of spatio-temporal variance in the high-dimensional space. As used herein, the term "manifold" refers to a space of a given dimensionality, such that high-dimensional video data may be projected into a low-dimensional space, where videos in a same class are located close to one another in the low-dimensional space. A spatial-temporal-historical consistency (STHC) model trained in a source domain may thereby be adapted encouraging it to handle spatio-temporal variance by learning such low-dimensional manifolds.

[0018] It can be challenging to learn reliable manifold structures without label supervision. To mitigate this, spatial and temporal data augmentation may be performed. For each unlabeled target video, spatial and temporal augmentations may be applied in a stochastic manner to produce augmented videos that outline the neighboring structure of the manifold that the video resides on. Encouraging consistent classification predictions for the video and its augmented versions ensures that they are drawn from the same spatio-temporal manifold. With these learned manifolds, the model can generalize to the unlabeled target domain and can make consistent predictions for videos drawn from the same manifold.

[0019] The models described herein may be used to implement, for example, action recognition within a video stream. Such action recognition may be used for a variety of applications, such as sports analysis and training, industrial training, and health and rehabilitation. For example, video may be taken of a patient undergoing physical therapy and action recognition may be used to identify the patient's form and whether they are performing an exercise correctly. This information may then be used to inform medical decision making, for example with a healthcare professional making changes to the patient's physical therapy and rehabilitation regimen.

[0020] Referring now to FIG. 1, an exemplary video stream 100 is illustrated, being made up of a series of sequential video frames 101. Each video frame 101 may include an image that captures a camera's view at a particular time. Included in the video stream 100 is a series of action frames 102 that show a particular action taking place. The action frames 102 have a beginning time 104 and an ending time 106.

[0021] The video stream 100 may include labels that explicitly indicate the beginning time 104 and the ending time 106 of an action, and may further include a label that identifies the action taking place in the action frames 102. In an unannotated video stream, the video stream 100 may lack labels for these time markers and classes.

[0022] Each frame 101 includes multiple kinds of information. For example, each frame 101 may include appearance information that includes RGB values for each pixel of the frame 101, or any other appropriate encoding of static visual information. In addition, more than one frame 101 may be considered together to identify motion information, for example by identifying optical flow information that captures dynamic and motion information from one frame 101 to the next.

[0023] In one example of a video stream 100, the video stream 100 may capture a sporting event, such as a long jump. The action frames 102 may show the actual jump taking place, while frames 101 before and after the action frames 102 may show related or unrelated activity, such as the jumper's run leading up to the jump and the jumper landing and walking away.

[0024] Source-free video domain adaptation (SFVDA) may be derived from UVDA, sharing the same goal of learning an adaptive video classification model H using a source dataset $S = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_M, y_M)\}$ of M labeled videos $\mathbf{X}$ having respective labels y and a target dataset $T = \{\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N\}$ of N unlabeled videos. In SFVDA, H is learned in two steps, with a pertaining step learning weights using the

source domain data S in a supervised fashion and with an adaptation step that adapts the pretrained wights to the target domain using T in an unsupervised fashion. Separating S from the adaptation process prevents the labeled data from being disclosed, enabling privacy-safe adaptation.

[0025] The model H may be decoupled as $H = F \circ G \circ C$, where F, G, and C are a frame feature extractor, a temporal feature extractor, and a classifier, respectively. For each video $(\mathbf{X}, y) \in S$, segment-wise frame sampling may be performed, with $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ being the m frames of the video. The frames may be evenly divided into K segments, and one sample from each segment may be taken. This produces a K-frame video snippet of class y $(\overline{\mathbf{X}}, y)$, where $|\overline{\mathbf{X}}| = K$. Using $\overline{\mathbf{X}}$ as input to F produces a sequence of frame features, which are then processed by G to learn temporal information. Average pooling of all frame features may be performed to get a vector representation for $\overline{\mathbf{X}}$, which may be used as input to the classifier C to calculate a cross-entropy loss with y and to update H with stochastic gradient descent and back-propagation.

[0026] Given a model H with pretrained weights learned from the labeled source data S, the STHC model adapts the weights to the target domain using unlabeled data T. This is accomplished by learning spatial-temporal manifolds that videos from a same category are mapped to, despite spatio-temporal variance. The model H can thereby surpass the domain gap between the source domain and the target domain to achieve better generalization performance. To learn such manifolds, a number of spatially or temporally augmented videos are generated for each target video using stochastic augmentation. Consistent predictions are enforced for the video and its augmented versions from spatial, temporal, and historical points of view.

[0027] Referring now to FIG. 2, a block diagram of a STHC model framework is shown. An unlabeled video $\mathbf{U} \in T$, having N frames, is sampled into k parts by block 202, with one frame being randomly selected from each part to produce a K-frame sequence $\overline{\mathbf{U}} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_K\}$. The sequence is processed by spatial augmentation 204 to generate spatially augmented video and by temporal augmentation 206 to generate temporally augmented video. The unlabeled video 202, the output of spatial augmentation 204, and the output of temporal augmentation 206 are each independently processed by a pre-trained STHC model 208.

[0028] Per-frame spatial augmentation 204 may be performed to generate $\overline{\mathbf{U}}_s = \{\psi(\mathbf{u}_1), \psi(\mathbf{u}_2), \ldots, \psi(\mathbf{u}_N)\}$, where ψ is a stochastic spatial augmentation function, implying that different augmented frames can be obtained if ψ is applied to a same frame at different times. The stochastic function can thereby generate potentially infinite augmented videos around $\overline{\mathbf{U}}$ in the low-dimensional manifold space, which enables more precise modeling of the manifold.

[0029] In some embodiments, the spatial augmentation 204 may include a random combination of image transformations, which may be selected from a pool that includes color inversion, translation, and adjustment. In some embodiments, the spatial augmentation 204 may include a removal of image information, for example by replacing a random square patch of a frame with gray pixels.

[0030] The spatially augmented video sequence $\overline{\mathbf{U}}_s$ may be spatially adjacent with $\overline{\mathbf{U}}$ in the low-dimensional manifold space, as it is generated from $\overline{\mathbf{U}}_s$ with frame content perturbed. The semantic category information, such as the actions that are shown in the video, should be preserved.

[0031] A spatial proximity constraint may be enforced to ensure spatial consistency in block 210, for example by minimizing the discrepancy between the classification predictions of $\overline{\mathbf{U}}_s$ and $\overline{\mathbf{U}}$ as:

$$\mathcal{L}_s = s\big(H(\overline{\mathbf{U}}), H(\overline{\mathbf{U}}_s)\big)$$

where $s(\cdot,\cdot)$ is a prediction consistency measurement function. Exemplary consistency measurement functions include cross-entropy, Kullback-Leibler divergence, and cosine similarity. The Kullback-Leibler divergence may be used in particular, but should not be considered limiting.

[0032] A temporal proximity constraint may similarly be enforced to ensure temporal consistency in block 212. For the sampled frame sequence $\overline{\mathbf{U}}$, stochastic temporal augmentation 206 may be applied to generate $\overline{\mathbf{U}}_t = \{\phi(\mathbf{u}_1), \phi(\mathbf{u}_2), \ldots, \phi(\mathbf{u}_K)\}$, where ϕ denotes a stochastic temporal function that drops $\mathbf{u}_k$ out of the sequence, for example at a rate of 0.5. This makes $\overline{\mathbf{U}}_t$ a sparser version of $\overline{\mathbf{U}}$. Despite having fewer frames, it is expected that $\overline{\mathbf{U}}_t$ will preserve the motion dynamics of $\overline{\mathbf{U}}$ in the low-dimensional manifold space, which helps learn the neighboring manifold structure of $\overline{\mathbf{U}}$. As $\overline{\mathbf{U}}_t$ may be viewed as a harder version of $\overline{\mathbf{U}}$, with different motion dynamics, enforcing prediction consistency between them encourages the model to surpass the motion dynamics variations between domains and to facilitate model adaptation.

[0033] As with spatial consistency 210, temporal proximity may be enforced in block 212 by minimizing the discrepancy between the classification predictions $\overline{\mathbf{U}}$ and $\overline{\mathbf{U}}_t$ as:

$$\mathcal{L}_T = s\big(H(\overline{\mathbf{U}}), H(\overline{\mathbf{U}}_t)\big)$$

where $s(\cdot,\cdot)$ is the same affinity function as was used in the spatial consistency 210.

[0034] Historical consistency 214 is used to reinforce the temporal consistency 212. Given that $\overline{\mathbf{U}}$ is randomly sampled from the k video segments, the temporally augmented $\overline{\mathbf{U}}_t$ is generated to encourage temporal consistency by masking some frames

to make a hard augmented version. However, when the video is long or the segment number k is small, the variance within each segment may be large, which the temporal consistency described above does not take into account.

[0035] In each training round, the past sequences can be viewed as other versions of the current sequence, and consistency can be enforced between past predictions and the current prediction. A memory bank M may be used to store past Q predictions for each video. M may be used to calculate the historical consistency loss as:

$$\mathcal{L}_h = \mathbb{E}_{\mathbf{p}_u \sim M}\left[s\left(\mathbf{p}_u, H(\bar{\mathbf{U}})\right)\right]$$

where $\mathbf{p}_u$ is a historical prediction for a video clip sampled from $\bar{\mathbf{U}}$. The value of Q should be kept small to prevent predictions from obsolete models from being stored, as enforcing consistency with old knowledge would prevent the model from learning new knowledge. An exemplary value of $Q = 2$ reaches generalizable results.

[0036] During training, a learning objective may be formulated as

$$\mathcal{L} = \mathcal{L}_{im} + \alpha\left(\mathbb{E}_{\bar{\mathbf{U}} \sim T}(\mathcal{L}_s + \mathcal{L}_t + \mathcal{L}_h)\right)$$

where α is a hyper-parameter and $\mathcal{L}_{im}$ is an information maximization loss, such as:

$$\mathcal{L}_{im} = -\mathbb{E}_{\bar{\mathbf{U}} \sim T} \sum_{r=1}^{R} H(\bar{\mathbf{U}}) \log H(\bar{\mathbf{U}}) + \sum_{r=1}^{R} p_r \log(p_r)$$

where $\mathbf{p} = -\mathbb{E}_{\bar{\mathbf{U}} \sim T} H(\bar{\mathbf{U}})$ is the mean of predictions over all target videos, and where p_r is the r^{th} dimension of $\mathbf{p}$. The first term of $\mathcal{L}_{im}$ minimizes the entropy of the probability, encouraging the model to make confident predictions, while the second term maximizes the entropy of $\mathbf{p}$ to encourage the samples to be balanced over all classes.

[0037] Referring now to FIG. 3, pseudo-code for adapting the parameters of an STHC model is shown, following the framework shown in FIG. 2.

[0038] Referring now to FIG. 4, a method for training the STHC model is shown. The model is trained in block 400 in a two-stage process. Block 402 begins by pre-

training the model according to a labeled training dataset in a source domain. This training may be performed using supervised training of, for example, an action classifier that identifies an action performed in an input video.

[0039] Block 404 then adapts the model according to an unlabeled training dataset that may be in a target domain. For example, the target domain may include video taken under different lighting conditions or from a different viewpoint. By enforcing spatial, temporal, and historical consistency across different augmented versions of the videos in the unlabeled training dataset, the generalizability of the model can be increased while maintaining its efficacy in the source domain.

[0040] After the model is trained in block 400, the model may be deployed 410 for use, for example in a healthcare facility. New video may be recorded 420 and actions may be classified 430 in the new video relating to recorded motion of a person in the video. Block 440 generates a report relating to the recognized action(s).

[0041] The STHC model can be readily extended to other video domain adaptation problems. In partial domain adaptation, classes in the target domain may be a subset of classes in the source domain. Under a source-free constraint, this implies that only samples from a part of all the classes of the source domain are used to adapt the model. The STHC model can be directly applied to such a setting, as the consistency terms described above have no assumption on the class distributions of the target domain. The class-balancing term of $\mathcal{L}_{im}$ may be dropped in the partial domain adaptation setting.

[0042] In open-set domain adaptation, source classes may be a subset of the target classes. The entropy of predictions can be used as a measurement of uncertainty to divide target samples into groups, for example using K-means clustering. The group with higher mean uncertainty may be regarded as samples from unknown classes and may be rejected for adaptation.

[0043] In black box model adaptation, the source model may not be available for adaptation, but may instead serve as a black box that produces outputs for given inputs. The STHC model may be extended to this scenario with a two-stage solution. In a first stage, the black box model may be treated as a teacher model to train a student model of randomly initialized weights via knowledge distillation on target data. In the second stage, the student model may be adapted as above.

[0044] Referring now to FIG. 5, a diagram of action recognition is shown in the context of a healthcare facility 500. A patient may perform rehabilitation exercises 506 at the healthcare facility 500, where their activity may be recorded with one or more video cameras. The rehabilitation exercises may make use of treatment systems 504, such as exercise equipment and biometric monitoring devices.

[0045] The healthcare facility may include one or more medical professionals 502 who provide information relating to the patient's rehabilitation progress and information provided by the treatment systems 504. As action recognition 508 is performed on the recorded rehabilitation exercises 506, information may be automatically generated regarding the patient's form. Ineffective or hazardous may be identified and a report may be generated for a medical professional 502 to use in decision making. For example, based on the report, the medical professional 502 may correct the patient's motion to aid in their further rehabilitation.

[0046] The different elements of the healthcare facility 500 may communicate with one another via a network 510, for example using any appropriate wired or wireless communications protocol and medium. The medical professional 502 may thereby receive information relating to the rehabilitation exercises 506 from the action recognition 508.

[0047] Referring now to FIG. 6, additional detail on the STFC model 208 is shown. The model 208 includes a frame feature extractor 602 that generates features of each frame of an input video. The frame feature extractor 602 may include any appropriate image feature encoding model, such as a residual neural network (ResNet).

[0048] Temporal information encoder 604 captures information about the motion of elements within the video over time, and a classifier 606 uses this information to identify, for example, an action that is performed within the video. The temporal information encoder 604 takes input as a sequence of frame features, extracted by frame feature extractor 602, that represent the whole video. The temporal information encoder 604 may be implemented as a multi-layer perceptron (MLP) that models the interactions of temporal information among all the frames, with a mean pooling operator over all the frame features.

[0049] The classifier 606 takes the video feature as input and outputs a prediction score for the category that the video belongs to. The classifier 606 may be implemented as a single fully-connected layer.

[0050] Referring now to FIG. 7, an exemplary computing device 700 is shown, in accordance with an embodiment of the present invention. The computing device 700 is configured to perform action recognition.

[0051] The computing device 700 may be embodied as any type of computation or computer device capable of performing the functions described herein, including, without limitation, a computer, a server, a rack based server, a blade server, a workstation, a desktop computer, a laptop computer, a notebook computer, a tablet computer, a mobile computing device, a wearable computing device, a network appliance, a web appliance, a distributed computing system, a processor-based system, and/or a consumer electronic device. Additionally or alternatively, the computing

device 700 may be embodied as one or more compute sleds, memory sleds, or other racks, sleds, computing chassis, or other components of a physically disaggregated computing device.

[0052] As shown in FIG. 7, the computing device 700 illustratively includes the processor 710, an input/output subsystem 720, a memory 730, a data storage device 740, and a communication subsystem 750, and/or other components and devices commonly found in a server or similar computing device. The computing device 700 may include other or additional components, such as those commonly found in a server computer (e.g., various input/output devices), in other embodiments. Additionally, in some embodiments, one or more of the illustrative components may be incorporated in, or otherwise form a portion of, another component. For example, the memory 730, or portions thereof, may be incorporated in the processor 710 in some embodiments.

[0053] The processor 710 may be embodied as any type of processor capable of performing the functions described herein. The processor 710 may be embodied as a single processor, multiple processors, a Central Processing Unit(s) (CPU(s)), a Graphics Processing Unit(s) (GPU(s)), a single or multi-core processor(s), a digital signal processor(s), a microcontroller(s), or other processor(s) or processing/controlling circuit(s).

[0054] The memory 730 may be embodied as any type of volatile or non-volatile memory or data storage capable of performing the functions described herein. In operation, the memory 730 may store various data and software used during operation of the computing device 700, such as operating systems, applications, programs, libraries, and drivers. The memory 730 is communicatively coupled to the processor 710 via the I/O subsystem 720, which may be embodied as circuitry and/or components to facilitate input/output operations with the processor 710, the memory 730, and other

components of the computing device 700. For example, the I/O subsystem 720 may be embodied as, or otherwise include, memory controller hubs, input/output control hubs, platform controller hubs, integrated control circuitry, firmware devices, communication links (e.g., point-to-point links, bus links, wires, cables, light guides, printed circuit board traces, etc.), and/or other components and subsystems to facilitate the input/output operations. In some embodiments, the I/O subsystem 720 may form a portion of a system-on-a-chip (SOC) and be incorporated, along with the processor 710, the memory 730, and other components of the computing device 700, on a single integrated circuit chip.

[0055] The data storage device 740 may be embodied as any type of device or devices configured for short-term or long-term storage of data such as, for example, memory devices and circuits, memory cards, hard disk drives, solid state drives, or other data storage devices. The data storage device 740 can store program code 740A for training a model, 740B for recognizing actions within a video, and/or 740C for generating a report on the recognized actions. Any or all of these program code blocks may be included in a given computing system. The communication subsystem 750 of the computing device 700 may be embodied as any network interface controller or other communication circuit, device, or collection thereof, capable of enabling communications between the computing device 700 and other remote devices over a network. The communication subsystem 750 may be configured to use any one or more communication technology (e.g., wired or wireless communications) and associated protocols (e.g., Ethernet, InfiniBand®, Bluetooth®, Wi-Fi®, WiMAX, etc.) to effect such communication.

[0056] As shown, the computing device 700 may also include one or more peripheral devices 760. The peripheral devices 760 may include any number of additional

input/output devices, interface devices, and/or other peripheral devices. For example, in some embodiments, the peripheral devices 760 may include a display, touch screen, graphics circuitry, keyboard, mouse, speaker system, microphone, network interface, and/or other input/output devices, interface devices, and/or peripheral devices.

[0057] Of course, the computing device 700 may also include other elements (not shown), as readily contemplated by one of skill in the art, as well as omit certain elements. For example, various other sensors, input devices, and/or output devices can be included in computing device 700, depending upon the particular implementation of the same, as readily understood by one of ordinary skill in the art. For example, various types of wireless and/or wired input and/or output devices can be used. Moreover, additional processors, controllers, memories, and so forth, in various configurations can also be utilized. These and other variations of the processing system 700 are readily contemplated by one of ordinary skill in the art given the teachings of the present invention provided herein.

[0058] Referring now to FIGs. 8 and 9, exemplary neural network architectures are shown, which may be used to implement parts of the present models, such as the STHC model 208. A neural network is a generalized system that improves its functioning and accuracy through exposure to additional empirical data. The neural network becomes trained by exposure to the empirical data. During training, the neural network stores and adjusts a plurality of weights that are applied to the incoming empirical data. By applying the adjusted weights to the data, the data can be identified as belonging to a particular predefined class from a set of classes or a probability that the input data belongs to each of the classes can be output.

[0059] The empirical data, also known as training data, from a set of examples can be formatted as a string of values and fed into the input of the neural network. Each

example may be associated with a known result or output. Each example can be represented as a pair, (x, y), where x represents the input data and y represents the known output. The input data may include a variety of different data types, and may include multiple distinct values. The network can have one input node for each value making up the example's input data, and a separate weight can be applied to each input value. The input data can, for example, be formatted as a vector, an array, or a string depending on the architecture of the neural network being constructed and trained.

[0060] The neural network "learns" by comparing the neural network output generated from the input data to the known values of the examples, and adjusting the stored weights to minimize the differences between the output values and the known values. The adjustments may be made to the stored weights through back propagation, where the effect of the weights on the output values may be determined by calculating the mathematical gradient and adjusting the weights in a manner that shifts the output towards a minimum difference. This optimization, referred to as a gradient descent approach, is a non-limiting example of how training may be performed. A subset of examples with known values that were not used for training can be used to test and validate the accuracy of the neural network.

[0061] During operation, the trained neural network can be used on new data that was not previously used in training or validation through generalization. The adjusted weights of the neural network can be applied to the new data, where the weights estimate a function developed from the training examples. The parameters of the estimated function which are captured by the weights are based on statistical inference.

[0062] In layered neural networks, nodes are arranged in the form of layers. An exemplary simple neural network has an input layer 820 of source nodes 822, and a single computation layer 830 having one or more computation nodes 832 that also act

as output nodes, where there is a single computation node 832 for each possible category into which the input example could be classified. An input layer 820 can have a number of source nodes 822 equal to the number of data values 812 in the input data 810. The data values 812 in the input data 810 can be represented as a column vector. Each computation node 832 in the computation layer 830 generates a linear combination of weighted values from the input data 810 fed into input nodes 820, and applies a non-linear activation function that is differentiable to the sum. The exemplary simple neural network can perform classification on linearly separable examples (e.g., patterns).

[0063] A deep neural network, such as a multilayer perceptron, can have an input layer 820 of source nodes 822, one or more computation layer(s) 830 having one or more computation nodes 832, and an output layer 840, where there is a single output node 842 for each possible category into which the input example could be classified. An input layer 820 can have a number of source nodes 822 equal to the number of data values 812 in the input data 810. The computation nodes 832 in the computation layer(s) 830 can also be referred to as hidden layers, because they are between the source nodes 822 and output node(s) 842 and are not directly observed. Each node 832, 842 in a computation layer generates a linear combination of weighted values from the values output from the nodes in a previous layer, and applies a non-linear activation function that is differentiable over the range of the linear combination. The weights applied to the value from each previous node can be denoted, for example, by w_1, w_2, ... w_{n-1}, w_n. The output layer provides the overall response of the network to the input data. A deep neural network can be fully connected, where each node in a computational layer is connected to all other nodes in the previous layer, or may have

other configurations of connections between layers. If links between nodes are missing, the network is referred to as partially connected.

[0064] Training a deep neural network can involve two phases, a forward phase where the weights of each node are fixed and the input propagates through the network, and a backwards phase where an error value is propagated backwards through the network and weight values are updated.

[0065] The computation nodes 832 in the one or more computation (hidden) layer(s) 830 perform a nonlinear transformation on the input data 812 that generates a feature space. The classes or categories may be more easily separated in the feature space than in the original data space.

[0066] Embodiments described herein may be entirely hardware, entirely software or including both hardware and software elements. In a preferred embodiment, the present invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

[0067] Embodiments may include a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. A computer-usable or computer readable medium may include any apparatus that stores, communicates, propagates, or transports the program for use by or in connection with the instruction execution system, apparatus, or device. The medium can be magnetic, optical, electronic, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. The medium may include a computer-readable storage medium such as a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk, etc.

[0068] Each computer program may be tangibly stored in a machine-readable storage media or device (e.g., program memory or magnetic disk) readable by a general or special purpose programmable computer, for configuring and controlling operation of a computer when the storage media or device is read by the computer to perform the procedures described herein. The inventive system may also be considered to be embodied in a computer-readable storage medium, configured with a computer program, where the storage medium so configured causes a computer to operate in a specific and predefined manner to perform the functions described herein.

[0069] A data processing system suitable for storing and/or executing program code may include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code to reduce the number of times code is retrieved from bulk storage during execution. Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) may be coupled to the system either directly or through intervening I/O controllers.

[0070] Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

[0071] As employed herein, the term “hardware processor subsystem” or “hardware processor” can refer to a processor, memory, software or combinations thereof that cooperate to perform one or more specific tasks. In useful embodiments, the hardware processor subsystem can include one or more data processing elements (e.g., logic

circuits, processing circuits, instruction execution devices, etc.). The one or more data processing elements can be included in a central processing unit, a graphics processing unit, and/or a separate processor- or computing element-based controller (e.g., logic gates, etc.). The hardware processor subsystem can include one or more on-board memories (e.g., caches, dedicated memory arrays, read only memory, etc.). In some embodiments, the hardware processor subsystem can include one or more memories that can be on or off board or that can be dedicated for use by the hardware processor subsystem (e.g., ROM, RAM, basic input/output system (BIOS), etc.).

[0072] In some embodiments, the hardware processor subsystem can include and execute one or more software elements. The one or more software elements can include an operating system and/or one or more applications and/or specific code to achieve a specified result.

[0073] In other embodiments, the hardware processor subsystem can include dedicated, specialized circuitry that performs one or more electronic processing functions to achieve a specified result. Such circuitry can include one or more application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), and/or programmable logic arrays (PLAs).

[0074] These and other variations of a hardware processor subsystem are also contemplated in accordance with embodiments of the present invention.

[0075] Reference in the specification to "one embodiment" or "an embodiment" of the present invention, as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, the appearances of the phrase "in one embodiment" or "in an embodiment", as well any other variations, appearing in various places throughout the specification are not

necessarily all referring to the same embodiment. However, it is to be appreciated that features of one or more embodiments can be combined given the teachings of the present invention provided herein.

[0076] It is to be appreciated that the use of any of the following "/", "and/or", and "at least one of", for example, in the cases of "A/B", "A and/or B" and "at least one of A and B", is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of "A, B, and/or C" and "at least one of A, B, and C", such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended for as many items listed.

[0077] The foregoing is to be understood as being in every respect illustrative and exemplary, but not restrictive, and the scope of the invention disclosed herein is not to be determined from the Detailed Description, but rather from the claims as interpreted according to the full breadth permitted by the patent laws. It is to be understood that the embodiments shown and described herein are only illustrative of the present invention and that those skilled in the art may implement various modifications without departing from the scope and spirit of the invention. Those skilled in the art could implement various other feature combinations without departing from the scope and spirit of the invention. Having thus described aspects of the invention, with the details

and particularity required by the patent laws, what is claimed and desired protected by Letters Patent is set forth in the appended claims.

WHAT IS CLAIMED IS:

1. A method for training a model, comprising:

performing spatial augmentation (204) on an unlabeled input video to generate spatially augmented video;

performing temporal augmentation (206) on the input video to generate temporally augmented video;

generating predictions (208), using a model that was pre-trained on a labeled dataset, for the unlabeled input video, the spatially augmented video, and the temporally augmented video; and

adapting parameters (404) of the model using the predictions while enforcing temporal consistency, temporal consistency, and historical consistency.

2. The method of claim 1, wherein enforcing spatial consistency includes comparing the spatially augmented video to the unlabeled input video.

3. The method of claim 1, wherein enforcing temporal consistency includes comparing the temporally augmented video to the unlabeled input video.

4. The method of claim 1, wherein enforcing historical consistency includes comparing the prediction for the unlabeled input video to a prediction of a second unlabeled video used in a prior training iteration.

5. The method of claim 1, wherein generating predictions includes performing action recognition.

6. The method of claim 5, further comprising identifying an action in treatment video relating to a rehabilitation activity of a patient and generating a report for healthcare professionals to use in decision making for rehabilitative care of the patient.

7. The method of claim 1, further comprising pre-training the model using supervised learning with a labeled dataset.

8. The method of claim 7, wherein a domain of the labeled dataset is distinct from a domain of the unlabeled dataset.

9. The method of claim 1, wherein adapting parameters of the model further includes an information maximization loss.

10. The method of claim 9, wherein the information maximization loss includes a first term that minimizes an entropy of probability and a second term that maximizes an entropy of a mean of predictions over videos in a target domain.

11. A system for training a model, comprising:

a hardware processor (710); and

a memory (740) that stores a computer program which, when executed by the hardware processor, causes the hardware processor to:

perform spatial augmentation (204) on an unlabeled input video to generate spatially augmented video;

perform temporal augmentation (206) on the input video to generate temporally augmented video;

generate predictions (208), using a model that was pre-trained on a labeled dataset, for the unlabeled input video, the spatially augmented video, and the temporally augmented video; and

adapt parameters (404) of the model using the predictions while enforcing temporal consistency, temporal consistency, and historical consistency.

12. The system of claim 11, wherein the computer program further causes the hardware processor to compare the spatially augmented video to the unlabeled input video.

13. The system of claim 11, wherein the computer program further causes the hardware processor to compare the temporally augmented video to the unlabeled input video.

14. The system of claim 11, wherein the computer program further causes the hardware processor to compare the prediction for the unlabeled input video to a prediction of a second unlabeled video used in a prior training iteration.

15. The system of claim 11, wherein the generated predictions include action recognition.

16. The system of claim 15, wherein the computer program further causes the hardware processor to identify an action in treatment video relating to a rehabilitation activity of a patient and to generate a report for healthcare professionals to use in decision making for rehabilitative care of the patient.

17. The system of claim 11, wherein the computer program further causes the hardware processor to pre-train the model using supervised learning with a labeled dataset.

18. The system of claim 17, wherein a domain of the labeled dataset is distinct from a domain of the unlabeled dataset.

19. The system of claim 11, wherein the computer program further causes the hardware processor to adapt the parameters with an information maximization loss.

20. The system of claim 19, wherein the information maximization loss includes a first term that minimizes an entropy of probability and a second term that maximizes an entropy of a mean of predictions over videos in a target domain.

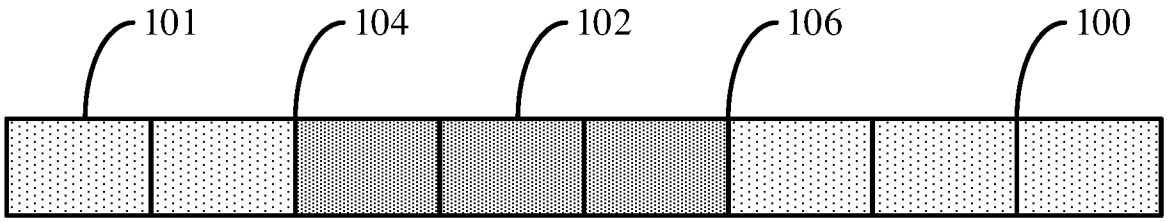

101
104
102
106
100

FIG. 1

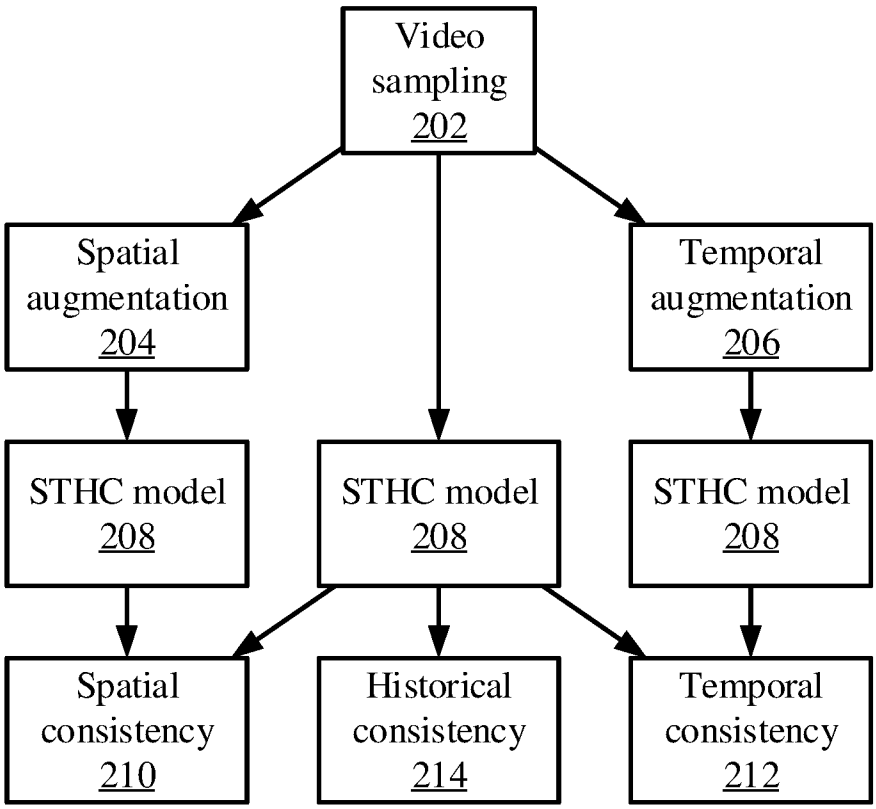

Video sampling 202
Spatial augmentation 204
Temporal augmentation 206
STHC model 208
STHC model 208
STHC model 208
Spatial consistency 210
Historical consistency 214
Temporal consistency 212

FIG. 2

Initialize an empty memory bank $\mathcal{M}$ for storing predictions for all training videos.
while not done **do**
  Randomly sample $\mathbf{U} \sim \mathcal{T}$.
  Get a sequence of frames $\bar{\mathbf{U}}$ from $\mathbf{U}$ with segment-wise random sampling.
  Get $\bar{\mathbf{U}}_s$ from $\bar{\mathbf{U}}$ by spatial augment.
  Get $\bar{\mathbf{U}}_t$ from $\bar{\mathbf{U}}$ by temporal augment.
  Calculate predictions $H(\bar{\mathbf{U}})$, $H(\bar{\mathbf{U}}_s)$, and $H(\bar{\mathbf{U}}_t)$.
  Calculate spatial consistency loss $\mathscr{L}_s$ with $H(\bar{\mathbf{U}})$, $H(\bar{\mathbf{U}}_s)$.
  Calculate temporal consistency loss $\mathscr{L}_t$ with $H(\bar{\mathbf{U}})$, $H(\bar{\mathbf{U}}_t)$.
  if $\mathcal{M}$ is not empty for $\mathbf{U}$ **then**
    Calculate historical consistency loss $\mathscr{L}_h$ with $H(\bar{\mathbf{U}})$ and $\mathcal{M}$.
  end if
  Calculate IM loss $\mathscr{L}_{im}$.
  Update $\mathcal{M}$ with $H(\bar{\mathbf{U}})$ with the first-in-first-out principle.
  Update H with the overall loss.
end while

FIG. 3

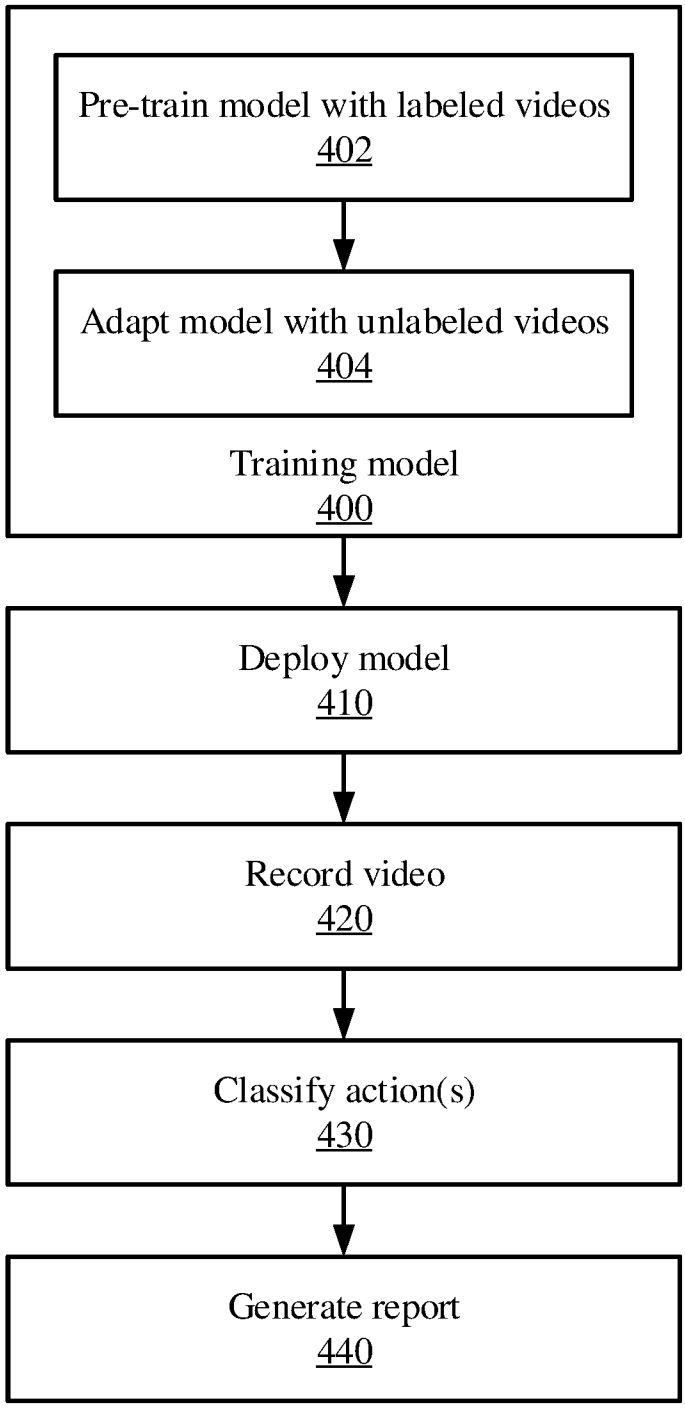

Pre-train model with labeled videos
402
Adapt model with unlabeled videos
404
Training model
400
Deploy model
410
Record video
420
Classify action(s)
430
Generate report
440

FIG. 4

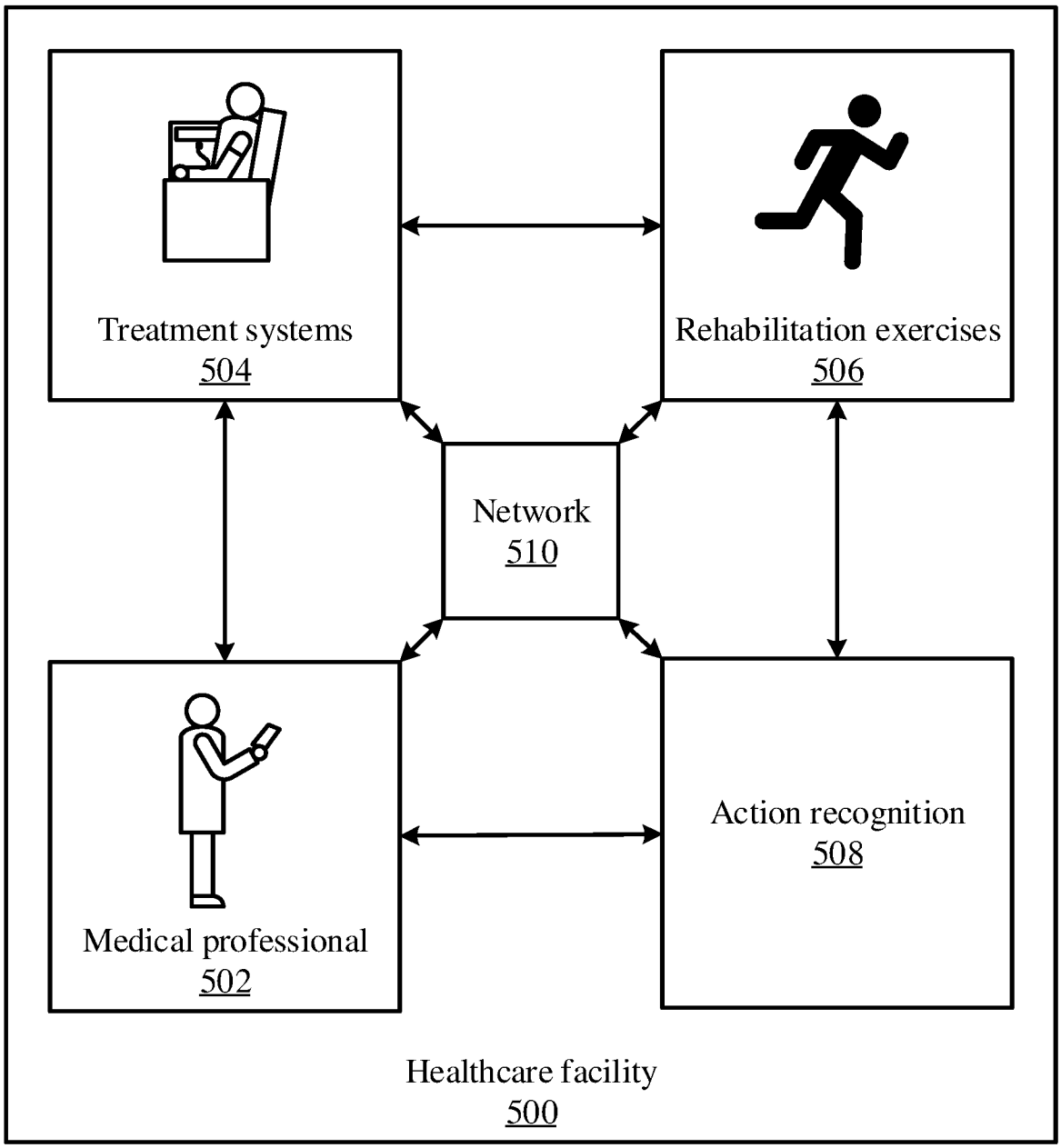

Treatment systems
504
Rehabilitation exercises
506
Network
510
Medical professional
502
Action recognition
508
Healthcare facility
500

FIG. 5

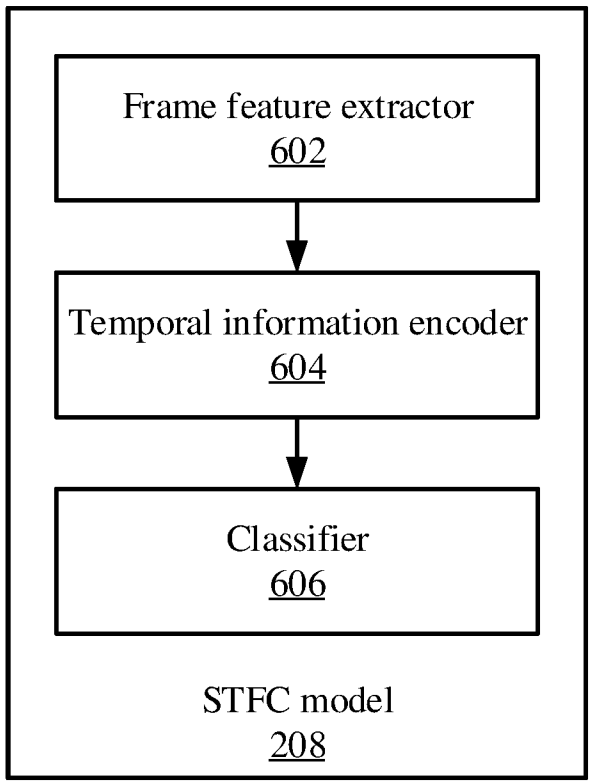

Frame feature extractor
602
Temporal information encoder
604
Classifier
606
STFC model
208

FIG. 6

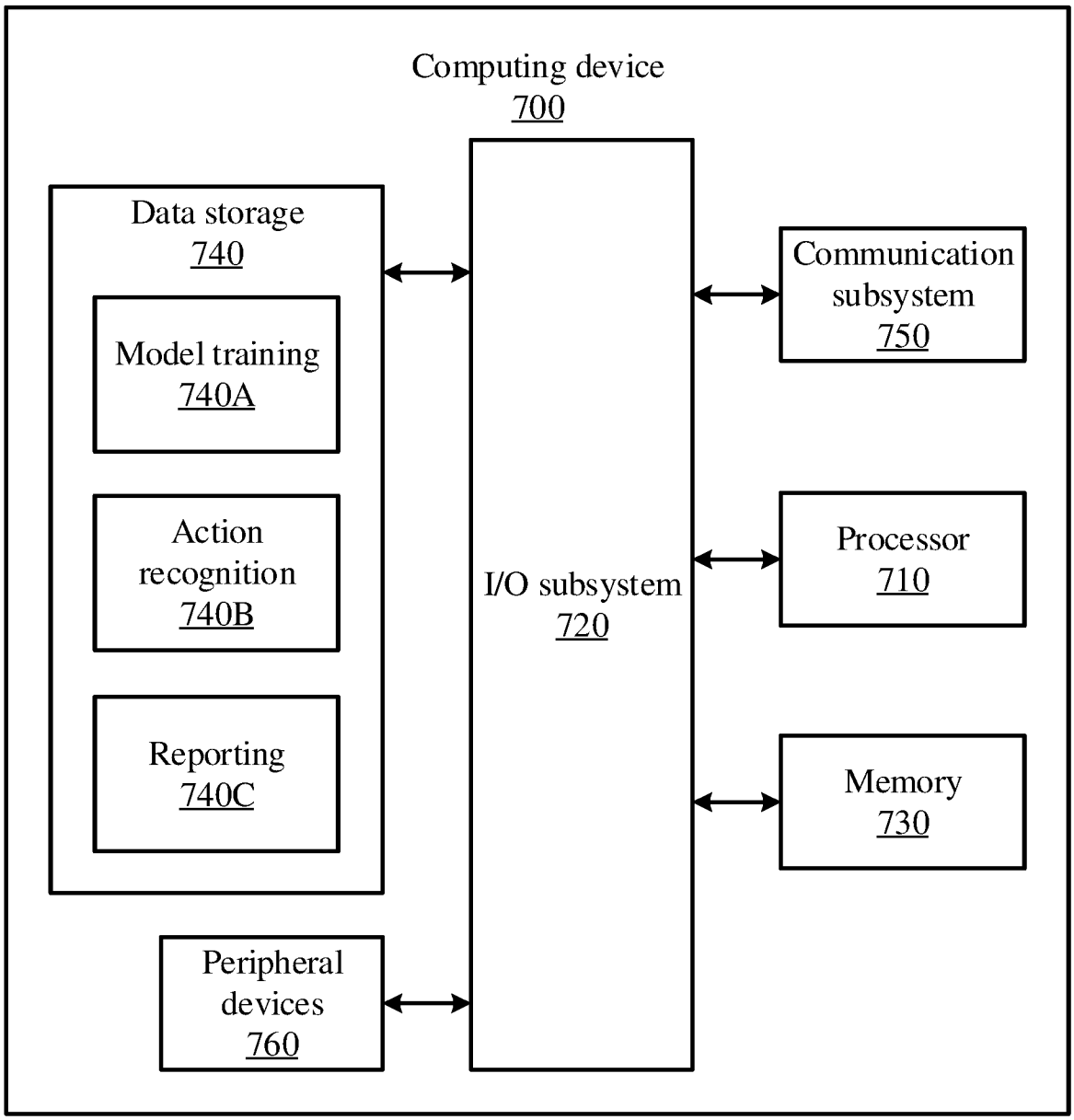

Computing device
700
Data storage
740
Model training
740A
Action recognition
740B
Reporting
740C
Peripheral devices
760
I/O subsystem
720
Communication subsystem
750
Processor
710
Memory
730

FIG. 7

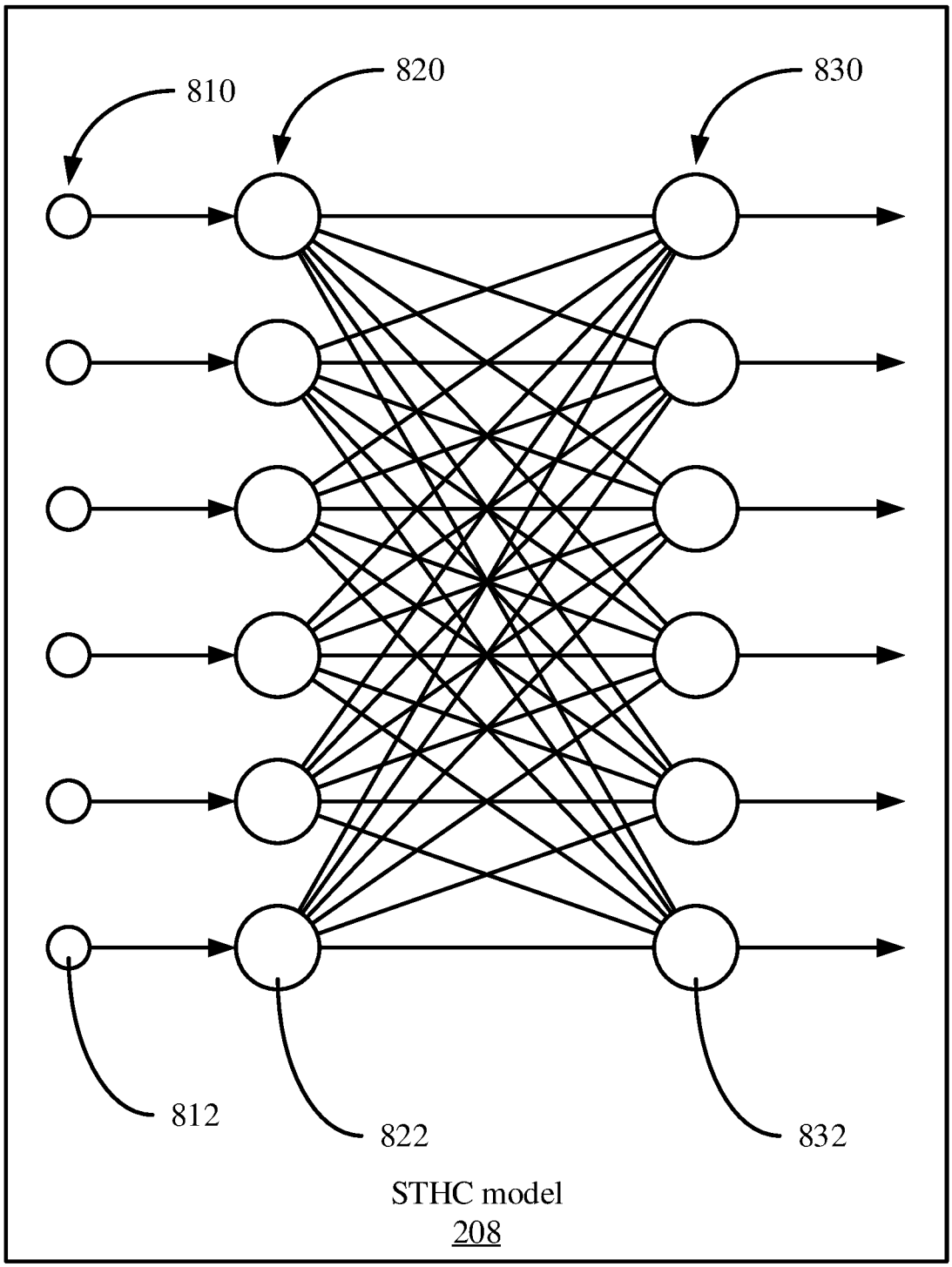


FIG. 8

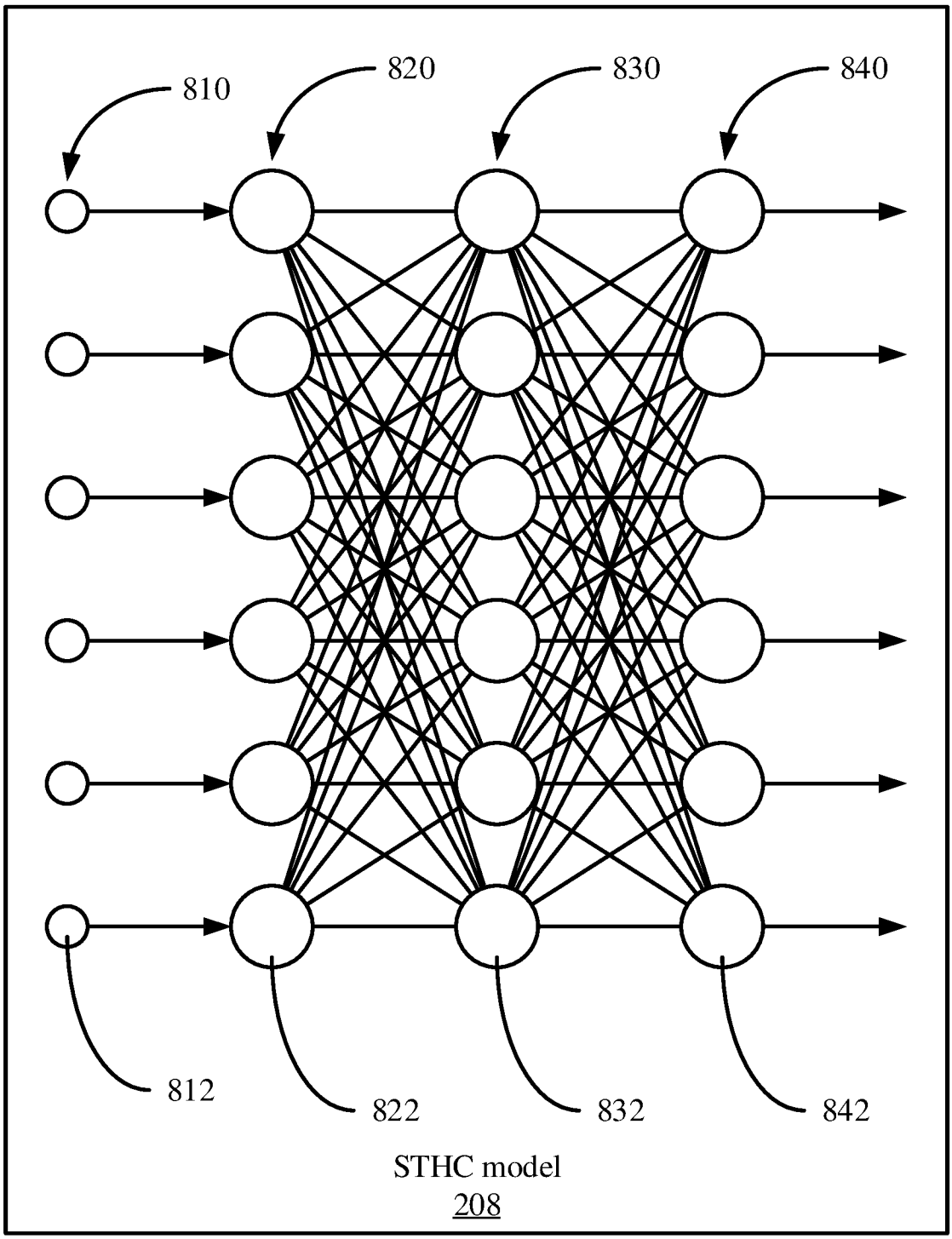


FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2023/037076

A. CLASSIFICATION OF SUBJECT MATTER

G06V 40/20(2022.01)i; **G06V 10/82**(2022.01)i; **G06V 10/764**(2022.01)i; **G16H 20/30**(2018.01)i; **G06N 3/096**(2023.01)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)

G06V 40/20(2022.01); G06K 9/62(2006.01); G06N 3/08(2006.01); G06Q 10/08(2012.01); G06V 10/82(2022.01); G07C 5/00(2006.01); G08B 29/18(2006.01)

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

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: training, augmentation, unlabeled input video, spatio-temporal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2022-106302 A1 (BAYER AKTIENGESELLSCHAFT) 27 May 2022 (2022-05-27) claim 1 and figure 1	1-20
A	US 2018-0053401 A1 (RAPIDSOS, INC.) 22 February 2018 (2018-02-22) claims 1, 15 and figures 1-2	1-20
A	US 2022-0156593 A1 (SALESFORCE.COM, INC.) 19 May 2022 (2022-05-19) claim 1 and figures 1-4	1-20
A	WO 2022-125930 A1 (GOOGLE LLC) 16 June 2022 (2022-06-16) claims 1, 4 and figures 1A-1D	1-20
A	US 2022-0188577 A1 (FRACTAL ANALYTICS PRIVATE LIMITED) 16 June 2022 (2022-06-16) claim 1 and figures 1-5	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	Date of mailing of the international search report
07 March 2024	**08 March 2024**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office **189 Cheongsa-ro, Seo-gu, Daejeon** **35208, Republic of Korea**	**LEE Kang Ha**
Facsimile No. **+82-42-481-8578**	Telephone No. **+82-42-481-5003**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2023/037076

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
WO 2022-106302 A1	27 May 2022	CN 116490903 A	25 July 2023
		EP 4248356 A1	27 September 2023
		US 2024-0005650 A1	04 January 2024
US 2018-0053401 A1	22 February 2018	US 10861320 B2	08 December 2020
		US 11790766 B2	17 October 2023
		US 2021-0233388 A1	29 July 2021
		WO 2018-039142 A1	01 March 2018
US 2022-0156593 A1	19 May 2022	None	
WO 2022-125930 A1	16 June 2022	EP 4260295 A1	18 October 2023
		US 2023-0260652 A1	17 August 2023
US 2022-0188577 A1	16 June 2022	None	