US 20250265463A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2025/0265463 A1**

Brown et al. (43) **Pub. Date: Aug. 21, 2025**

(54) **METHOD AND SYSTEM FOR SYMMETRIC RECOGNITION OF HANDED ACTIVITIES**

(71) Applicant: **Hinge Health, Inc.**, San Francisco, CA (US)

(72) Inventors: **Colin Brown**, Saskatoon (CA); **Andrey Tolstikhin**, Montreal (CA)

(73) Assignee: **Hinge Health, Inc.**, San Francisco, CA (US)

(21) Appl. No.: **19/197,703**

(22) Filed: **May 2, 2025**

Related U.S. Application Data

(63) Continuation of application No. 18/311,809, filed on May 3, 2023, now Pat. No. 12,314,855, which is a continuation of application No. 17/593,270, filed on Sep. 14, 2021, filed as application No. PCT/IB2020/052249 on Mar. 12, 2020, now Pat. No. 11,657,281.

(30) **Foreign Application Priority Data**

Mar. 15, 2019 (CA) 3036836

Publication Classification

(51) **Int. Cl.**

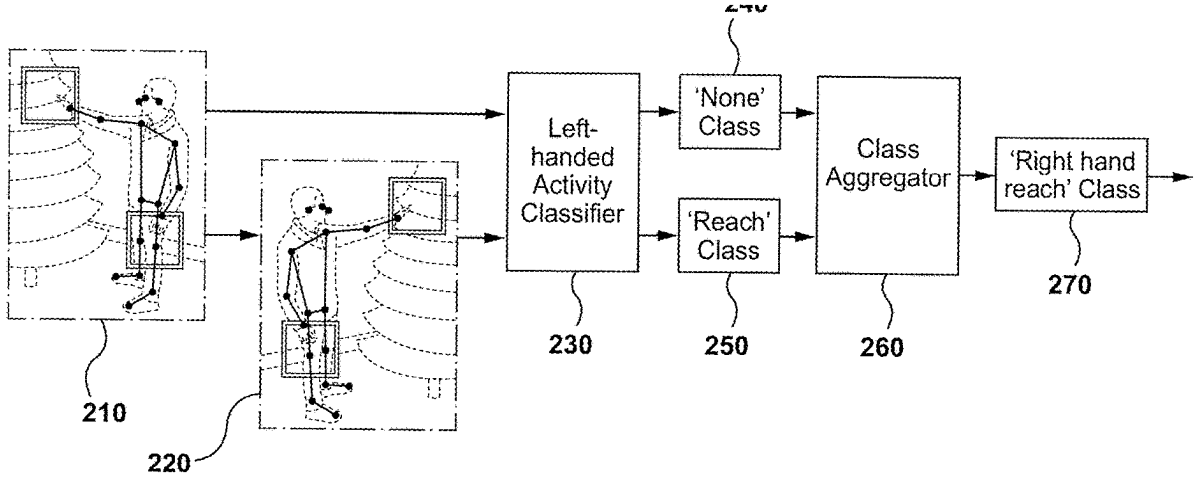
G06N 3/08	(2023.01)
G06V 10/774	(2022.01)
G06V 10/82	(2022.01)
G06V 40/10	(2022.01)
G06V 40/20	(2022.01)

(52) **U.S. Cl.**

CPC ***G06N 3/08*** (2013.01); ***G06V 10/7747*** (2022.01); ***G06V 10/82*** (2022.01); ***G06V 40/107*** (2022.01); ***G06V 40/20*** (2022.01)

(57) **ABSTRACT**

This disclosure describes an activity recognition system for asymmetric (e.g., left- and right-handed) activities that leverages the symmetry intrinsic to most human and animal bodies. Specifically, described is 1) a human activity recognition system that only recognizes handed activities but is inferenced twice, once with input flipped, to identify both left- and right-handed activities and 2) a training method for learning-based implementations of the aforementioned system that flips all training instances (and associated labels) to appear left-handed and in doing so, balances the training dataset between left- and right-handed activities.



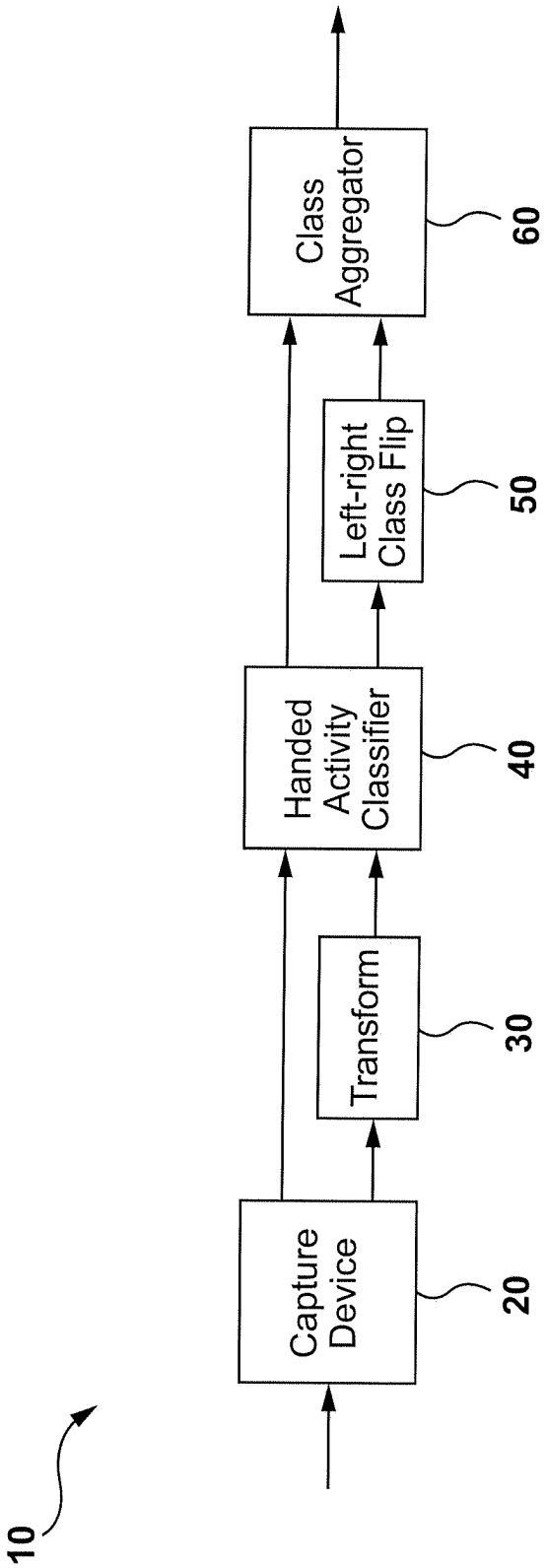

10
Capture Device
20
Transform
30
Handed Activity Classifier
40
Left-right Class Flip
50
Class Aggregator
60

FIG. 1

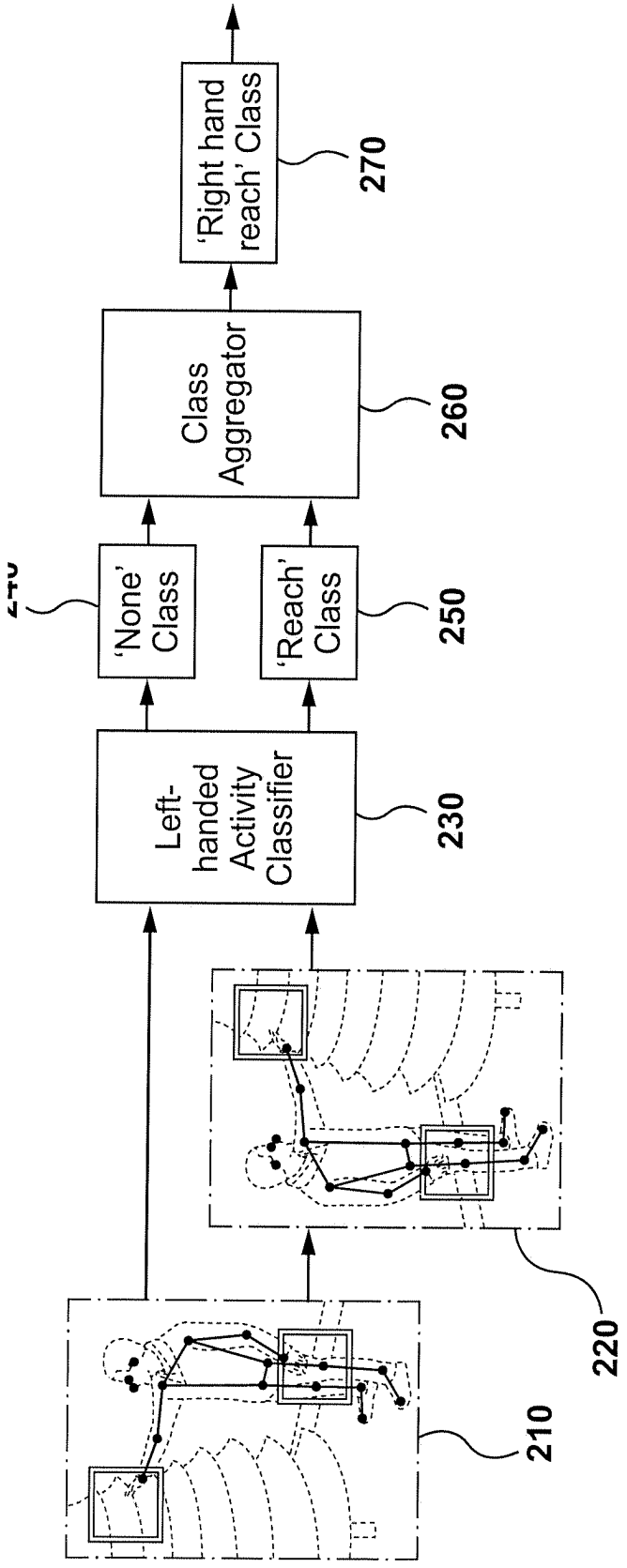

'Right hand reach' Class
270
Class Aggregator
260
'None' Class
'Reach' Class
250
Left-handed Activity Classifier
230
220
210

FIG. 2

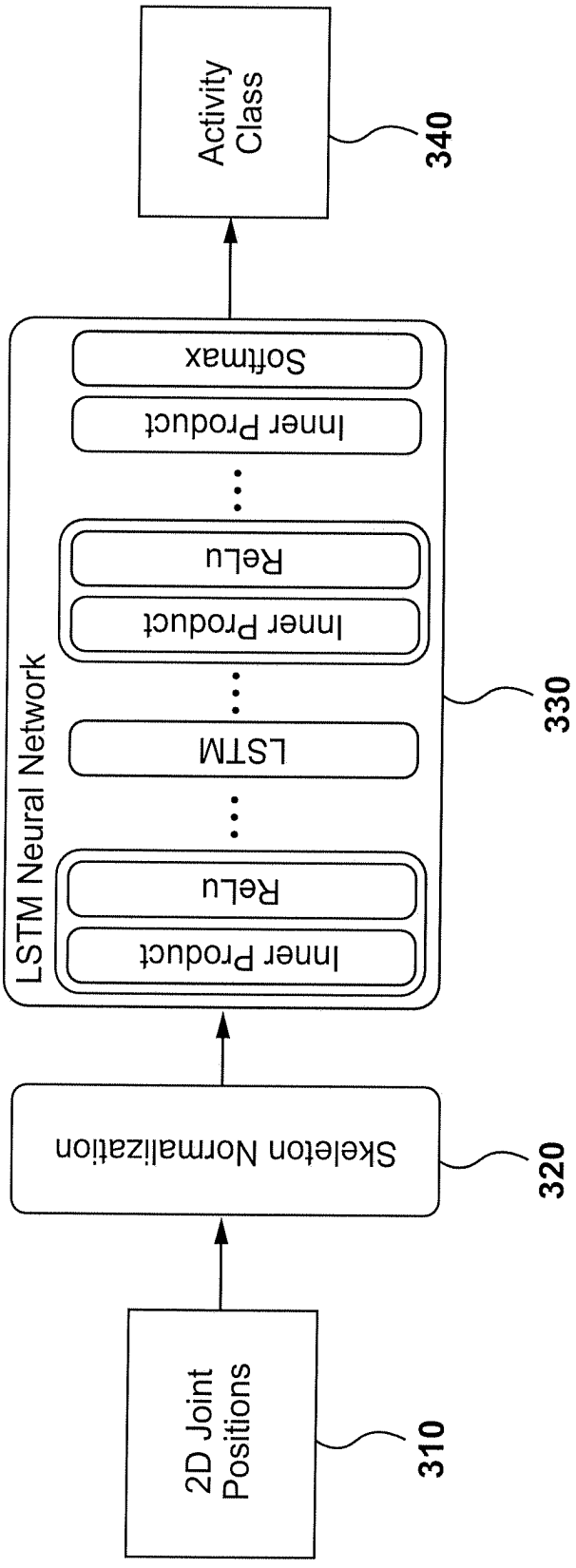

2D Joint Positions
310
Skeleton Normalization
320
LSTM Neural Network
Inner Product
ReLu
LSTM
Inner Product
ReLu
Inner Product
Softmax
330
Activity Class
340

FIG. 3

METHOD AND SYSTEM FOR SYMMETRIC RECOGNITION OF HANDED ACTIVITIES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. application Ser. No. 18/311,809, filed May 3, 2023, which is a continuation of U.S. application Ser. No. 17/593,270, filed Sep. 14, 2021, now U.S. Pat. No. 11,657,281, which is a national stage entry of International Application No. PCT/IB2020/052249, filed Mar. 12, 2020, which claims priority to Canadian Application No. 3,036,836, filed Mar. 15, 2019, each of which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] This disclosure relates to systems and methods for computer-based recognition of activities. Specifically, it relates to recognition of those activities that are not left-right symmetric.

BACKGROUND

[0003] Deep learning-based methods have been proposed that detect human activities in video data (e.g., waving, running). Many human activities are left-right asymmetric (e.g., shaking hands, kicking a ball) but may be performed equally on either side of the body. There exist a broad range of applications for activity recognition methods that can differentiate between left- and right-handed executions of a given activity and consider each as a separate class. For example, in a soccer analytics application, it may be important to distinguish between a player shooting with their left foot or right foot.

[0004] Furthermore, this kind of activity recognition may be more broadly applicable to non-human animals, robots or symmetric actors of any kind. For example, in animal research, it may be useful to have a system that can distinguish asymmetric activities performed by an animal predominantly on its left or right side.

[0005] However, requiring an activity recognition system to distinguish between left- and right-handed activities effectively doubles the number of classes that may be predicted, which may increase the cost and complexity of the activity recognition system as well as the training database and training procedure for learning-based systems.

[0006] An additional challenge that arises for learning-based systems stems from the imbalanced population distribution of persons who are predominantly left-handed versus those who are predominantly right-handed. Trained learning-based classifiers can be sensitive to the distribution of classes in the database. However, it is common for large training datasets to be sampled from public data and it is known that a vast majority of the human population is right-handed. Thus, while it may be desirable for an activity recognition to be equally accurate for left- and right-handed activities, it may be challenging to acquire a training dataset with equal numbers of left- and right-handed activity examples, especially if the dataset is sampled from public data.

[0007] It is therefore desirable for an improvement in recognizing symmetric and non-symmetric activities.

SUMMARY

[0008] This disclosure describes an activity recognition system for left-right asymmetric (e.g., left- and right-handed) activities that leverages the left-right mirror symmetry intrinsic to most human and animal bodies. Specifically, described is 1) an activity recognition system that only recognizes activities of one handedness (e.g., only left handed activities) but is inferenced twice, once with input flipped horizontally, to identify both left- and right-handed activities and 2) a training method for learning-based implementations of the aforementioned system that flips all training instances (and associated labels) to appear left-handed and in doing so, balances the training dataset between left- and right-handed activities.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] In drawings which illustrate by way of example only a preferred embodiment of the disclosure,

[0010] FIG. **1** is a representation of an activity recognition system as a block diagram.

[0011] FIG. **2** is a schematic representation of an embodiment of a use case of the activity recognition system.

[0012] FIG. **3** is a schematic representation of an embodiment of a left-handed activity classifier.

DETAILED DESCRIPTION

Activity Recognition System

[0013] The activity recognition system **10**, may comprise a capture device **20**, a transform module **30** to flip input from this capture device horizontally, a handed activity classifier **40**, a module to perform a left-right flip on the activity class output from the left-right activity classifier **50**. The activity recognition system may also contain a class aggregator component **60** to determine the final activity class or classes given information from both flipped and non-flipped input.

[0014] An activity class is a named activity or plurality of related activities (e.g., 'waving' which may denote waving at one of a variety of different speeds and/or in one of many different fashions) by some actors (e.g., person, animal) at some temporal scale (e.g., still image of an activity being performed, short gestures like winks, long activities like golfing). An activity class such as 'none,' 'default' or 'idle' may represent the absence of a relevant activity. The plurality of valid activity classes and the representation of an activity class may depend on the application of the activity recognition system and on the specific embodiment of that system.

[0015] An activity class may be associated with a handedness (left or right) if it is typically performed left-right asymmetrically in some way. Such an activity need not be performed exclusively with one hand nor with one side of the body but may be labelled appropriately as left- or right-handed such as by the dominant or leading side of the body, to distinguish it from its symmetric counterpart. Activity classes representing activities that are performed with left-right symmetry may be considered as special cases of asymmetric activity classes in which the left- and right-handed activity classes denote the same activity or plurality of activities. For example, head nodding may be an activity that does not require left-right activity classes. Similarly, for some applications of a classifier it may not matter whether the activity is left- or right-handed. For example, for 'car

driving,' it may not matter if the driver is holding the steering wheel with the right or left hand or both hands.

[0016] Data from a capture device **20** may be provided to the handed activity classifier **40** from which an activity class or plurality of class probabilities may be inferred and then may be passed to a class aggregator module to determine the outputted activity class or classes **60**. Data from the capture device may also be provided to the transform module **30**, such as a horizontal flip. The activity data from the capture device **20** may be sent simultaneous, or sequentially, in either order, to the horizontal flip transform module and the handed activity classifier **40**. The transform module **30** may perform a left-right flip transform on the input data before the handed activity classifier **40** infers a handed activity class or plurality of class probabilities. The activity class (or classes) from the flipped input data may then be left-right flipped **50** (i.e., from a left-handed class to a right-handed class) and may be provided as a second input to the class aggregator module **60**, which may determine a final activity class or plurality of classes given the two inputs.

[0017] Each of these components are described below in detail.

[0018] The capture device **20** may be embodied by a system or device for accessing, capturing or receiving activity data. For example, the activity data may comprise images, video files, live video data such as from a webcam, video streaming client, human pose estimation system, or a motion capture system. The activity data preferably is capable of representing persons or other relevant actors (e.g., animals, androids) in some spatially coherent manner. For example, the activity data may comprise one or more image frames of a video, image of depth values from a depth camera, and/or 2D or 3D skeleton/marker points. The activity data may be temporal, such as a sequence of frames or atemporal, such a single frame.

[0019] The transform module **30** may accept input data from the capture device and then perform a left-right (horizontal) flip transformation on that data. The exact form of the transformation may depend on the format of data provided by the capture device (e.g., image data versus skeleton data) but the result is the activity data is flipped horizontally so that events, or activities on the left side of the activity data are transformed to the right side and vice versa.

[0020] For example, the transform module **30** may accept a set of key-points representing skeletal joints of a targeted person as an array of X and Y coordinates, normalized in the range 0 to 1. In this case, the horizontal flip module may flip the X coordinates of each joint as 1-X and leaving the Y coordinates unaltered. For activity data that is video or images, the horizontal flip module may transform the data using a mirror image transformation.

[0021] In applications with other expected asymmetries and symmetries in the input and training data, other types of transformations may be used. For example, in some applications, vertical symmetry may be present and instead of a horizontal flip module, a vertical flip module may be used as a transform module. Similarly, for temporal activity data, the transform module may re-order the activity data in reverse time order if forward/backward time symmetry exists in the application. The transform module may do one or more transformations, and in some applications multiple transform modules may be used in parallel and sequence if the activity classes contain more activities. For example, using both a horizontal flip module and a vertical flip module, an activity may be inferred as left-handed and upside down.

[0022] The handed activity classifier **40** may accept flipped or non-flipped input activity data and output an inferred activity class or plurality of class probabilities of one particular handedness.

[0023] The handed activity classifier **40** may identify left-handed activities or may equally identify right-handed activities. As is noted above, a left-handed activity class may not necessarily represent activities performed exclusively with the left-hand but instead are some activities that are performed in an asymmetric way and are deemed as left-handed to distinguish them from their right-handed counterparts. For example, a left-handed activity classifier designed for analysis of tennis footage may recognize forehanded strokes only when it is viewed to be performed with the player's left hand and may otherwise output some non-activity class.

[0024] While a single handed activity classifier **40** is shown, in embodiments, there may be two handed activity classifiers, with a first classifier accepting the non-flipped input activity data and a second classifier accepting the flipped input activity data. If a single handed activity classifier is used, it may process both the flipped and non-flipped input activity data, simultaneously but separately, or sequentially to separately infer an activity class for each of the flipped input activity data and the non-flipped input activity data. In an embodiment, a handed classifier may be loaded into memory once and then the frame run through the classifier once unflipped and once flipped.

[0025] The handed activity classifier may be implemented by a classifier, such as a convolutional-neural-network-based classifier, decision tree, support-vector-machine, or hand-crafted classifier. The handed activity classifier may accept input data from the capture device and output one activity class or a plurality of class probabilities. The classifier may use temporal information if present, such as by using a recurrent-neural-network (RNN).

[0026] With reference to FIG. **3**, for example, the left-handed activity classifier may be implemented as a long-short-term memory (LSTM) RNN that accepts flipped and non-flipped joint-coordinates as input and comprises as series of fully connected layers, activation layers (e.g., rectified linear unit layers), LSTM layers and softmax layers.

[0027] With reference to FIG. **3**, in an embodiment, the left-handed activity classifier is implemented as a long-short term memory (LSTM) neural network that accepts 2D joint positions of a person as input, and outputs a one-hot vector representation of an estimated activity class. In other words, in the output vector, one element of the output vector corresponding to the estimated class has a high value. In this embodiment classifier, the input 2D joint positions may be first normalized into a [0,1] coordinate space **320** and then passed to a series of stacked inner-product layers, restricted linear unit (ReLu) layers and LSTM layers before being passed to a softmax layer, **330**, which computes the output class **340**. The inner product layers and LSTM layers may be parameterized by trainable weights that are learned by training on a database of labelled activity data.

[0028] The left-right class flip module **50** may accept an activity class or plurality of class probabilities of one particular handedness from the handed activity classifier inferred on flipped input data and may produce the corre-

sponding activity class or plurality of class probabilities of opposite handedness. As an example, if the left-handed activity class inferred by the left-handed activity classifier is "waving with left hand," the left-right class flip module **50** may produce "waving with right hand" to be passed to the class aggregator module **60**. While referred to as a left-right class flip module, the class flip module **50** may flip classes vertically, forward/back or some other transformation depending on the application and the transformations performed by the transform module.

[0029] The class aggregator module **60** may accept left-handed and right-handed activity classes or class probabilities from the result of the activity classifier inferring on flipped and non-flipped input data and may produce a single class or plurality of class probabilities (e.g., 'none' if no activity recognized by either input, a non-'none' class if recognized by either input and a single activity class chosen by some rule if both inputs are non-'none' or a combined class). As an example, if inputs to the class aggregator module **60** is "none" for the non-flipped input data and "waving with right hand" from the left-right class flip module **50**, then the class aggregator module **60** may produce "waving with right hand" as the output class. Similarly, if the input to the class aggregator module **60** is "waving with left hand" for the non-flipped input data and "none" from the left-right class flip module **50**, then the class aggregator module **60** may produce "waving with left hand" as the output class. If the input to the class aggregator module **60** is "waving with left hand" for the non-flipped input data and "waving with the right hand" for the flipped input data the class aggregator module **60** may produce a single aggregate class such as "waving with both hands" or a single class such as "waving with the right hand" if requirements of the use-case only require a single handed class to be identified or a plurality of classes such as "waving with left hand" and "waving with right hand."

[0030] Depending on the requirements of the use-case, in one embodiment, the class aggregator module **60** may instead produce a plurality of classes. For example, the class aggregator may only produce as output those input classes that are not "none," and so may output a varying number of outputs, depending on the input.

[0031] If the input to the class aggregator is a plurality of class probabilities from the handed activity classifier **40** and from the left-right flip class module **50**, in one embodiment, the output may be a single most probable class or a combined set of class probabilities of both left- and right-classes.

[0032] These described system modules may be separate software modules, separate hardware units or portions of one or more software or hardware components. For example, the software modules may consist of instructions written in a computer language such as the Python programming language, C++ or C #with suitable modules, such as created using software from Caffe, TensorFlow, or Torch, and run on computer hardware, such as a CPU, GPU or implemented on an FPGA. The system may be run on desktop, mobile phone or other platforms such as part of an embedded systems that includes suitable memory for holding the software and activity data. The system may be integrated with or connect to the capture device.

[0033] With reference to FIG. **2**, in an example, a frame from a video depicting a person reaching for an ornament on a tree **210** is provided by the capture device. Joints and bone connections (in blue) and bounding boxes around the person's hands (in fuchsia) have been overlaid onto the frame as examples of additional or alternative forms of input data that may be provided by the capture device. The original **210** and horizontally flipped **220** frame images are passed to a left-handed activity classifier **230**. The left-handed activity classifier **230** recognizes the 'reach' activity class **250** in the flipped input, where the left arm of the person appears to be reaching. With the original input, where the right arm appears to be reaching, the left-handed activity classifier **230** produces a 'none' activity class **240**. In this example, the class aggregator **260** determines that the resulting output class is 'right hand reach' given that a 'reach' activity was only detected in the flipped input.

Training Method

[0034] In the preferred case that the handed activity classifier **40** is implemented as some machine learning-based module, it may be parameterized by trainable weights that need to be trained on a class-labelled database of activity data. In this case, the training activity data is in the same format as to be provided by the capture device.

[0035] In this embodiment, the following training procedure may be used. For training instances with a class label representing a right-handed activity, the class label may be flipped to represent the corresponding left-handed activity label and the associated instance activity data, such as the skeleton data or video frame data, may also be flipped horizontally such that the right-handed activity appears as a left-handed activity. Performing these transforms produces a training dataset with only left-handed class labels and correctly corresponding instance activity data. The handed activity classifier **40** may then be trained on this transformed training dataset and may only learn to recognize left-handed activities.

[0036] The above procedure describes a way to create a dataset with only left-handed activities, but an equivalent procedure may be used to create a dataset with only right-handed activities. In either case, the handed activity classifier **40** may be trained on a dataset containing only example activities of a single handedness.

[0037] Training may be performed on a publicly available dataset (e.g., MPII, UFC101) or other training dataset (e.g., in-house captured and/or annotated dataset), with the back-propagation algorithm (or other appropriate learning algorithm) in an appropriate training framework such as Caffe (or TensorFlow, Torch, etc.) using appropriate hardware such as GPU hardware (or CPU hardware, FPGAs, etc.).

[0038] The embodiments of the invention described above are intended to be exemplary only. It will be apparent to those skilled in the art that variations and modifications may be made without departing from the disclosure. The disclosure includes all such variations and modifications as fall within the scope of the appended claims.

I/we claim:

1. A method comprising:

acquiring a set of keypoints that is representative of skeletal joints of a person, as derived from an analysis of a digital image that includes the person;

transforming the set of keypoints using a horizontal transformation, such that such that—

activities, if any, that are represented by the set of keypoints and that are performed on a left side are transformed to a right side in the transformed set of keypoints, and

activities, if any, that are represented by the set of keypoints and that are performed on the right side are transformed to the left side in the transformed set of keypoints;

applying, to the set of keypoints, a neural network to produce a first activity class for a given activity that is represented by the set of keypoints;

applying, to the transformed set of keypoints, the neural network to produce a second activity class for the given activity;

flipping the second activity class either from a left-handed class to a right-handed class or from the right-handed class to the left-handed class, so as to produce a third activity class having opposite handedness to the second activity class; and

outputting a predicted activity class for the given activity based on an analysis of the first and third activity classes.

2. The method of claim **1**, wherein the neural network is trained to identify either right-handed activities or left-handed activities, but not right- and left-handed activities.

3. The method of claim **1**, wherein the neural network is a long-short-term memory (LSTM) recurrent neural network that comprises a series of fully connected layers, activation layers, LSTM layers, and softmax layers.

4. The method of claim **1**,

wherein the set of keypoints represents the skeletal joints as X and Y coordinates, and

wherein the horizontal transformation causes each X coordinate to be reversed while leaving each Y coordinate unaltered.

5. A non-transitory medium with instructions stored thereon that, when executed by a processor, cause the processor to perform operations comprising:

transforming activity data that relates to a person performing an activity using a symmetric transformation, so as to create transformed activity data;

applying a neural network to the activity data and to the transformed activity data, so as to produce a first classification of the activity based on an analysis of the activity data and produce a second classification of the activity based on an analysis of the transformed activity data; and

outputting a predicted classification for the activity based on whether the first classification or an opposite handedness version of the second classification is dominant.

6. The non-transitory medium of claim **5**, wherein the operations further comprise:

flipping the second classification using a transformation that corresponds to the symmetric transformation, so as to produce the opposite handedness version of the second classification.

7. The non-transitory medium of claim **5**, wherein the activity data includes one or more digital images of the person performing the activity.

8. The non-transitory medium of claim **5**, wherein the activity includes a set of keypoints that correspond to different parts of the person.

9. The non-transitory medium of claim **8**, wherein the set of keypoints represents the different parts of the person as X and Y coordinates.

10. The non-transitory medium of claim **9**,

wherein the set of keypoints are normalized in a range of zero to one, and

wherein said transforming involves computing, for each X coordinate, an appropriate transformed value by subtracting that X coordinate from one.

11. The non-transitory medium of claim **5**, wherein the predicted classification is one of multiple predicted classifications output for the activity.

12. The non-transitory medium of claim **11**, wherein the multiple predicted classifications are representative of classifications for which there is some evidence in either the first classification or the second classification.

13. The non-transitory medium of claim **5**,

wherein the activity data is in temporal order and has forward/backward symmetry, and

wherein the symmetric transformation causes the activity data to be reordered in reverse temporal order.

14. The non-transitory medium of claim **5**, wherein the neural network is parameterized by trainable weights determined via analysis of a class-labeled training dataset.

15. A method for predicting a classification for an activity performed by an individual, the method comprising:

applying a neural network to an array that includes multiple pairs of X and Y coordinates, each of which is representative of a location of a corresponding one of multiple parts of the individual as the individual performs the activity, to produce a first classification of the activity;

applying the neural network to a symmetrically transformed version of the array to produce a second classification of the activity; and

outputting a predicted classification for the activity based on whether the first classification or an opposite handedness version of the second classification is dominant.

16. The method of claim **15**, further comprising:

transforming the array using a symmetrical transformation, such that—

activities, if any, that are performed on a left side and transformed to a right side, and

activities, if any, that are performed on the right side are transformed to the left side.

17. The method of claim **15**, wherein in the symmetrically transformed version of the array, each X coordinate is reversed while each Y coordinate is left unaltered.

18. The method of claim **15**, wherein in the symmetrically transformed version of the array, each Y coordinate is reversed while each X coordinate is left unaltered.

19. The method of claim **15**, wherein each pair of X and Y coordinates is indicative of two-dimensional (2D) position of the corresponding part of the individual in a digital image from which the multiple pairs of X and Y coordinates are derived.

20. The method of claim **15**, further comprising:

normalizing the multiple pairs of X and Y coordinates within a predetermined range defined by a lower bound and an upper bound.

* * * * *