**TITLE:** **SYSTEMS and METHODS for AUTOMATED BOXING DATA COLLECTION AND ANALYTICS PLATFORM**

**BACKGROUND OF THE INVENTION**

1. **TECHNICAL FIELD**
2. The present invention is directed to systems and methods for sports data collection, analytics and applications available as a service over a distributed network and remote users having access to a data and analytics platform.
3. The method receives an input and generates an output e.g. predicted output of athlete behaviour
4. **Description of the Prior Art**

Many currently available data capture methods are either (i) intrusive to the athletes performance or Collected manually by watching the event or event footage and entered into a database. Nearly all statistics are generated via human annotation where specialists use video play and pause function to manually collect raw statistics. Traditional video recording techniques have certain limitations, such as insufficient viewing angles, moving camera angles and zooms, non-calibrated images, and absence of tagged objects.

**DRAWINGS**

**BRIEF DESCRIPTION OF THE DRAWINGS**

**FIG 1** is an overall diagram of the method and interlinks with different components and elements of the system

**FIG 2** is a detailed flow diagram of the data collection and output process.

**FIG 3** is a flow diagram of an example process for classifying a new piece of data from the trained machine learning model.

**FIG 4** shows various end user access methods of data and interpretation

**FIG 5** is a an example of the boxing display data and analysis according to an exemplary embodiment

**FIG 6** is a detailed flow diagram of the training model.

**DETAIL DESCRIPTION**

(3) This specification describes how a method implemented as computer programs on one or more computers in one or more locations can determine specific punch classifications and categories to train a machine learning model to classify images, and, once trained, use the trained machine learning model to classify and process live video streams.

 (4) FIG. 2 shows the video processing and image classification system. The video processing and image classification system (200) is an example of a method implemented as a piece of Software on one or more computers in one or more locations in which the systems, components, and techniques described below are implemented.

(5) The video processing and image classification system (200)) uses a machine learning model to classify received images from live video feeds (201). For example, the video processing and image classification system can receive a new image and classify the new image to generate image classification data (207) that identifies one or more classification categories from a predetermined set of classification categories to which one or more actions depicted in the new image 205 belong. Once generated, the system 200 can store the live image classification data 209 in association with the new punch image 205 in a cloud based data repository (202), provide the punch image classification data 209 as input to another live video feed for further processing, or transmit the punch image classification data 209 to an end user of the system, e.g., transmit the image classification data 209 over IoT (Internet of Things) to a user device as depicted in **FIG4** (403).

(6) In particular, the embedding data 208 is maintained by the video processing and image classification system 200 in a cloud based secure replicated database (203), and is data that maps each object category in the set of object categories to a respective embedding of the object category in an embedding space.

(7) The video processing and image classification system is a model e.g. a deep convolutional neural network, that is configured to process input images to generate, for each input image, a predicted point based on a confidence ratio of >0.8 (or variants).

(8) To classify the new image 205, the video processing and image classification system 200 the new image 205 using the Visual Recognition model 207 to generate a predicted point classification for the new image. The system 200 then determines one or more classification embedding’s that are closest to the predicted point from among the classification embedding’s in the label embedding data 208 and classifies the new image 209 as including images of one or more objects that belong to the punch classification categories represented by the one or more closest classification embedding’s. Classifying new images is described in more detail below with reference to **FIG3**.

(9) To allow the visual recognition model 207 to be used to effectively classify punch input images, the system 200 includes a training engine 206 that receives training data 204 and uses the training data 204 to generate the classification embedding’s of the punch classification categories and to train the machine learning model 207.

(10) Generally, the training engine 206 generates the classification embedding’s such that a distance in the embedding space between the classification embedding’s for any two object categories reflects a degree of visual co-occurrence of the two object categories in images and then uses the generated embedding’s to train the machine learning model 207. Generating punch classification embedding’s and training a machine learning model is described in more detail below with reference to FIG. 3.

(11) FIG. 3 is a flow diagram of an example process 300 for training an image from a live video feed to classify images.

(12) For the purposes of clarity, the process 300 will be described as being performed by a system of one or more computers located in one or more locations. For example, an image classification system, e.g., the image classification system 200 of FIG. 2, appropriately programmed in accordance with this specification, can perform the process 300.

(13) The method receives training data for training a visual recognition model to classify punch images into categories (204).

(14) As described above, the method is a model (video processing and image classification system), e.g. a deep learning neural network, that is configured to receive an input image and to process the input image to generate a predicted classification in an embedding space in accordance with values of the parameters of the model.

(15) The training data (204) includes multiple training images of pre-defined classification and respective label data for each of the training images. The label data for a given training image identifies one or more object classifications from the set of object classifications to which one or more objects depicted in the training image belong. That is, the label data associates the training image with one or more of the object categories. These classifications are as follows:

1) Jab –Punch that is throw with the lead hand from a stance positive.

2) Cross – Straight punch thrown from the back hand from a stance position

3) Hook – Semi-circular punch that is aimed to land at the opponent’s side.

4) Uppercut – Punch rises from the bottom.

5) Overhand – Punch thrown with the back hand and travels over the head in a looping fashion.

(16) The system determines the classification for the object categories in the set of object categories (304). Once the classifications have been generated, the subsequent key frames reflects a degree of visual similarity of the two object categories in the training images. In particular, the degree of visual similarity is based on a relative frequency with which the same training image in the training data includes one or more objects that collectively belong to both of the two object categories, i.e. the relative frequency with which the label data for a training image associates both of the object categories with the training image. This is subsequently applied to the key extracted images from live video feed.

(17) To determine the classification for the object categories, the system determines matching information between each possible pair of object categories in the set of object categories as measured in the training data.

(18) For example, for a given pair of images, the matching information measure can be the confidence of the ratio between the probability that a training image includes one or more features that collectively belong to the key images extracted from the live video feed.The method then devises system for the mutual information measures which must satisfy:

**Live Data Feed Image Features = Confidence Ratio (Where confidence ratio >0.8 image is categorized and presented via UI)**

**Training Data Image Features**

(19) The method trains the machine learning recognition model on the training data to determine trained values of the model parameters from initial values of the model parameters (step 206).

(20) For each of the training images, the method processes the training image using the machine learning model in accordance with current values of the parameters of the visual recognition model to generate a predicted point in the classification for the training image.

(21) When there is more than one object classification is identified in the label data for the training image, the system can determine a combined embedding from the numeric embedding’s of the object categories identified in the label data for the training image. Once the method has trained the visual recognition model, the method can use the classifications and the trained parameter values to classify new images using the trained model.

(22) FIG. 3 is a flow diagram of an example process 300 for classifying a new image using a trained visual recognition model.

(23) For clarity the process 300 will be described as being performed by a system of one or more computers with access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet. For example, video processing and image classification system, e.g. video processing and image classification system 200 of Fig 2, appropriately applied in accordance with this specification, can perform the process 300.

(24) The system receives a new image to be classified (step 301).

(25) The method processes the newly split real-time video images using the trained visual recognition machine learning model to determine a confidence level point in the image (step 305). As described above, the visual recognition model has been configured through training to receive the newly split image from real time video and to process the new image to generate the classification based on confidence level in accordance with trained values of the parameters of the model.

(26) The method classifies the new image (from real-time video feed) as including images of one or more objects that belong to the object categories represented by the visual recognition model (step 305). Once the new image has been classified, the method can provide data identifying the object categories for presentation to the end user, e.g., punch classification ranked according to how close the corresponding features were to the predicted point, store data identifying the punch classification categories for later use, or provide the data identifying the punch classification to an external system for use for some immediate purpose.

(27) The herein description uses the term "configured" in connection with systems and computer software components. For a system of one or more computers to be configured to perform particular operations or actions means that the system has installed on it software, hardware, or a combination of them that in operation cause the system to perform the operations or actions. This can be hosted on-premises or via a cloud mechanism. For one or more computer programs to be configured to perform particular operations or actions means that the one or more programs include instructions that, when executed by a trigger, cause the infrastructure to perform the operations or actions.

(28) The herein description uses the term "Cloud” as a paradigm that enables ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned with minimal management effort over the Internet. This can optionally include code that creates an execution environment for computer programs, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

(27) Embodiments of the method described and the functional operations described in this description can be implemented in computer software, in computer hardware, including the structures disclosed in this description and their structural equivalents, or in combinations of one or more of them.

(28) The computer storage medium can be a machine-readable storage device, a machine-readable storage substrate, a random or serial access memory device, or a combination of one or more of them. Alternatively or in addition, the program instructions can be encoded on Cloud computing solution. The herein description includes Cloud Storage and Cloud Subsystem.

(29) A computer program, which may also be referred to or described as a program, software, a software application, an app, a module, a software module, a script, or code, can be written in any form of programming language, including compiled or interpreted languages, or declarative or procedural languages; and it can be deployed in any form, including as a standalone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A program may, but need not, correspond to a file in a file system. A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a WAN, LAN or across the internet.

(30) In this herein description, the term "database" (303) is used in the broad sense to refer to the existing collection of data: the data does not need to be structured in any particular way, or structured at all, and it can be stored on storage devices in one or more locations. For this method the storage is located via Cloud Storage.

(31) In this description the term "engine" is used in the broad sense to refer to a software-based system, subsystem, or process that is programmed to perform one or more specific functions. The training engine will be implemented as one or more software modules or components, installed on one or more computers in one or more locations. In some cases, one or more computers will be dedicated to a particular engine; in other cases, multiple engines can be installed and running on the same computer or computers. For this method is located via a Cloud based recognition engine.

(32) The processes and logic flows in this description can be performed by one or more programmable computers executing one or more computer programs to perform functions by operating on input data and generating output.

(33) Computers suitable for the execution of a computer program can be based on general or special purpose microprocessors or both, or any other kind of central processing unit. Generally, a central processing unit will receive instructions and data from a read only memory or a random access memory or both. The essential elements of a computer are a central processing unit for performing or executing instructions and one or more memory devices for storing instructions and data. A computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data. In this description data is received and stored in a cloud based system. A computer can also be embedded in another device e.g. Smart phone or tablet.

(34) Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices e.g. forms of removal storage media

(35) To provide for interaction with a user, embodiments of the subject matter identified in this description can be implemented on a computer having a display device for displaying information to the user and a keyboard and a pointing device, e.g. a mouse by which the user can provide input to the computer or touch screen. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's device in response to requests received from the web browser. Also, a computer can interact with a user by sending text messages or other forms of message to a personal device, e.g., a smartphone that is running a messaging application, and receiving responsive messages from the user in return.

(36) Data processing hardware for implementing Video processing and Image classification system can also include, for example, special-purpose hardware accelerator units (Graphical Processing Unit) for processing common and compute-intensive parts of machine learning training or production, i.e. workloads, data processing.

(37) Embodiments of the components of this description can be implemented in a computing system that includes a back end component, e.g., as a data server, or that includes a middleware component, e.g. an application server, or that includes a front end component, e.g., a client computer having a graphical user interface, a web browser, or an app through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such back end, middleware, or front end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network (LAN) and a wide area network (WAN), e.g., the Internet.

(38) The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. In some embodiments, a server transmits data, e.g., an HTML web page page, to a user device, e.g., for purposes of displaying data to and receiving user input from a user interacting with the device, which acts as a client. Data generated at the user device, e.g., a result of the user interaction, can be received at the server from the device.

(39) The components of this description contains many specific implementation details, however these should not be construed as limitations on the scope of any invention or on the scope of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. Certain features that are described in this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination.

(40) Similarly, while operations are depicted in the drawings and recited in the claims in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system modules and components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

(41) Particular embodiments of the subject matter have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results. As one example, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In some cases, multitasking and parallel processing may be advantageous.

**CLAIMS**

What is claimed is:

(1) A method comprising:

A computer-implemented method for analyzing activity of a fighter (Boxing or Combat Sports) from a sequence of images captured by real-time video feed based on:

Obtaining training data for training a visual recognition model having a fixed number of classification points based on punch categories.  
wherein the visual recognition machine learning model is configured to process input images to generate, for each input image, a predicted point in an embedding space, and  
wherein the training data comprises a multitude of training images and, for each training image, label data that identifies one or more object categories from a set of object categories to which one or more objects depicted in the training image belong;

processing the training image using the visual recognition machine learning model in accordance with current values of the parameters to generate a predicted point in the embedding space for the training image; and  
adjusting the current values of the parameters to reduce a distance between the predicted point in the embedding space and the numeric embedding’s of the object categories identified in the label data for the training image.

The live data feed images will be applied through the visual recognition model and identify, based on the training data and will identity the visual co-occurrence of the two images based on label data for the training image and live data to produce an output.

(2) The method of claim 1 , wherein the degree of visual co-occurrence is based on a relative frequency with which a same training image in the training data includes one or more objects that belong to one of the 5 punch categories that have been identified and applied to the live feed image.

(3) The method of any one of claims 1 or 2, wherein determining the respective classification categories comprises:  
determining a respective pointwise mutual information measure between each possible pair of object categories in the set of object categories as measured in the training data;  
constructing an identification of mutual information measures;  
performing an analysis of mutual information measures to determine an embedding;

(4)  The method of any one of claims 1 -3, wherein the visual recognition machine learning model is a deep convolutional neural network with ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned over the Internet.

(5) A method comprising:  
maintaining data that maps each punch classification in a set of categories to a respective numeric embedding of the object category in an embedding space, where in each piece of data reflects a degree of visual co-occurrence of the two object categories in images; receiving an input image;  
processing the input image using a visual recognition machine learning model, wherein the machine learning model has been configured to process the input image to generate a predicted label;  
determining, from the maintained data, one or more labels that are closest to the training data based on confidence score ; and  
classifying the live feed input image as including images of one or more objects that belong to the object categories represented by the one or more labels.

(6) The method of claim 5, wherein the machine learning model is a visual recognition machine learning model with ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned over the Internet.

(7) The method of any one of claims 5 or 6, wherein the degree of visual co-occurrence is based on a relative frequency with which a same training image in training data used to train the machine learning model includes one or more objects that collectively belong to both of the two object categories.

(8) The method of any one of claims 5-7, wherein determining, from the maintained data, one or more labels that are closest to the predicted point in the embedding space comprises:  
determining a predetermined number of labels that are closest to the predicted point in the embedding space.

(9) A method comprising one or more computers and one or more storage devices storing instructions that are operable, when executed by the one or more computers, to cause the one or more computers to perform the operations of the respective method of any one of claims 1 – 8. This includes ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned over the Internet.

(10). A computer storage medium encoded with instructions that, when executed by one or more computers, cause the one or more computers to perform the operations of the respective method of any one of claims 1 -8. This includes ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned over the Internet.

**Abstract**

A system is described for automated sports data collection and analytics. Different types of data, for example but not limited , movement data, punch count, punch classification are collected via video analysis in real time during a sports activity and transmitted to a cloud based platform together with other sports data including but not limited to timing, scoring, statistics, and events with a time code. The cloud based platform is optimised to compile correlate and organize various data related to the sports activity; store query and retrieve various live data and historical data and provide analytics and intelligence to different parties involved in a sports activity such as, but not limited to, Coaches, TV, Radio and Online Broadcasters, displays, viewers, social media and fans. These different parties may subscribe to licensed access to the cloud-based platform for customised real-time data feeds for their event\broadcast.











