Scientific Visualization, 2024, volume 16, number 2, pages 67 - 80, DOI: 10.26583/sv.16.2.06

Visualization and Classification of Human Movements Based on Skeletal Structure: A Neural Network Approach to Sport Exercise Analysis and Comparison of Methodologies

V.O. Kuzevanov[1], D. V. Tikhomirova[2]

National Research Nuclear University “MEPhI”, Moscow, Russia

[1] ORCID: 0009-0004-5415-1477, vl.kuzevanov@gmail.com
[2] ORCID: 0000-0002-0812-2331, dvsulim@mail.ru

Abstract

The authors of the paper review and compare different existing approaches to Human Action Recognition (HAR), analyze the advantages and disadvantages of platforms for extracting human skeletal structure from video stream, and evaluate the importance of visual representation in the motion analysis process. This paper presents an example implementation of one of the approaches to HAR based on the use of interpretability and visual expressiveness inherent in skeletal structures. In this work, an ad hoc network with Long Short-Term Memory (LSTM) for human activity classification is designed and implemented, which has been trained and tested in the domain of sports exercises. LSTM incorporation of memory cells and gating mechanisms not only mitigates the vanishing gradient problem but also enables LSTMs to selectively retain and utilize relevant information over extended sequences, making them highly effective in tasks with complex temporal dependencies. The problem with a fading gradient is quite common in deep neural networks and is that if the error is back propagated during the training of the network, the gradient can decrease strongly as it travels through the layers of the network to the initial layers. This can lead to the fact that the weights in the initial layers are practically not updated, which makes training of these layers impossible or slows down its process. The resulting solution can be used to create a real-time virtual fitness assistant. The resulting solution can be used to create a real-time virtual fitness assistant. In addition, this approach will make it possible to create interactive training applications with visualization of human skeletal structure, motion analysis and monitoring systems in the field of medicine and rehabilitation, as well as for the development of security systems with access control based on the analysis of visual data on the movement of human body parts.

Keywords: computer vision, neural network, machine learning, skeletal structure.

1. Introduction. Review of works in the field of action recognition

The problem of identifying human behavior patterns from video streams is challenging for computational devices, so one of the most important research objects in the scientific fields of computer vision and machine learning is the ability of computer systems to identify, segment, and classify human activities based on data collected by various sensors. Activity recognition information systems find applications in many fields, including video surveillance systems, human-computer interfaces, robotics, and healthcare. Such technologies can also be applied in sports, not only in broadcasting sporting events, but also in personal assistants and fitness assistants to improve the quality of an athlete's exercise performance. The vast possibility of application of such kind of systems and their usefulness determines the relevance of research

in this direction. One of the most popular approaches to the computation of complex tasks, allowing to surpass human capabilities, is the concept of deep learning, a subsection of machine learning (Deep Learning (DL) [1].

To date, there are many studies and techniques for recognizing human movements in video. Thus, M. Vrigkas, H. Nikou and I. A. Kakadiaris [2] propose the following decomposition of human actions (see Fig. 1) and hierarchy of recognition methods (see Fig. 2).

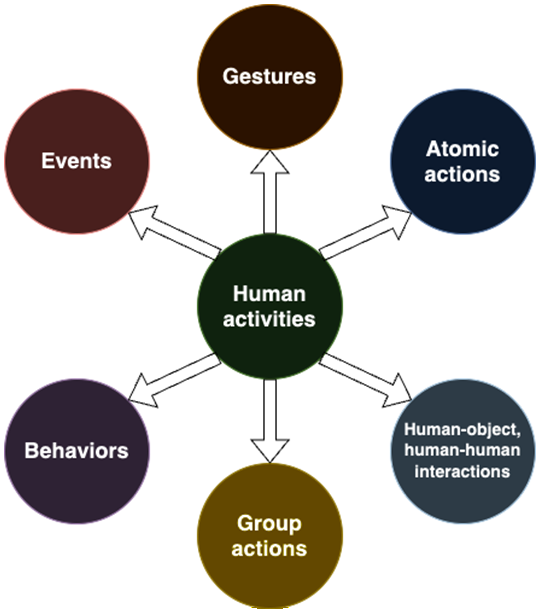

Figure 1. Decomposition of human actions

The goal of HAR is to learn and analyze activities from video sequences or still images. Recognition systems strive to correctly classify the input data into an underlying activity category. Based on complexity, human activities are categorized into 1) gestures, 2) atomic actions, 3) human-object or human-human interactions, 4) group actions, 5) behaviors, and 6) events.

Several approaches have been proposed in the field of human activity recognition research. They divide the research into 2D (with and without explicit shape models) and 3D approaches [3]. A new taxonomy has also been presented on human motion analysis, tracking from single and multi-view cameras and human activity recognition [4].

3D data modeling is also a new trend that has emerged with special cameras capable of detecting the depth of objects, which can be used for 3D reconstruction. The human body is composed of bones and connecting joints, allowing this structure to be modeled in 3D space using depth cameras, obtaining stronger features compared to modeling the human structure in 2D space. A study by Aggarwal J.K. and Xia L. [5] presents an established classification of HAR human activity recognition methods using 3D stereo and motion capture systems, focusing on methods that use depth data in their calculations. Systems like Microsoft Kinect [6] played an important role in motion capture by identifying skeletal structure and articulation motion using depth sensors.

Research distinguishes two main categories: (1) unimodal and (2) multimodal recognition methods according to the nature of the information coming from the sensors they use. They are further divided into subcategories according to how they model human activity.

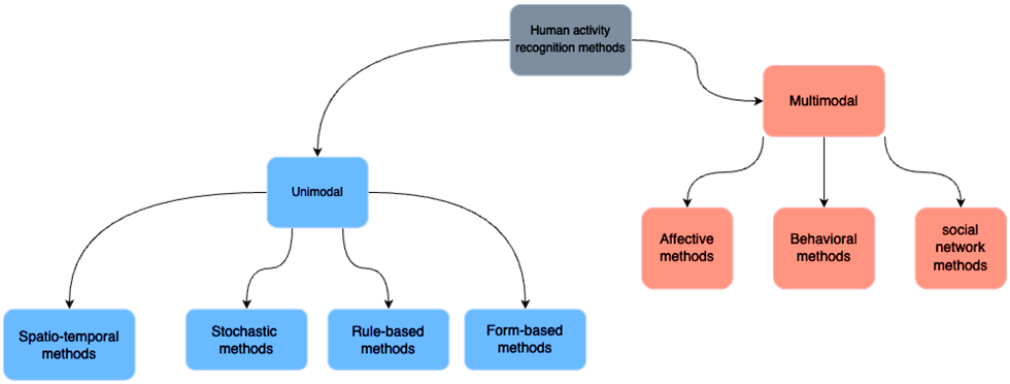

Figure 2. Hierarchical method categorization

Unimodal methods represent human activities based on single modality data (such as images) and they are further categorized into 1) spatio-temporal, 2) stochastic, 3) rule-based and 4) form-based methods.

Spatio-temporal methods are based on the concept of representing movements as a set of spatio-temporal features or trajectories [7]. Stochastic methods analyze human activity based on statistical models (e.g., hidden Markov models) [8]. Rule-based methods use a descriptive set of rules [9]. Shape-based methods use modeled shapes in space in the analysis of human movements [10].

Multimodal methods combine information from several modalities at once and fall into the following categories: 1) affective, 2) behavioral, and 3) social network methods [11].

Affective methods represent human activity through emotional communications and affective states [12]. Behavioral methods recognize various behavioral features, non-verbal multimodal cues such as gestures, facial expressions, and auditory cues [13]. Social network methods model the characteristics and behaviors of people at several levels of interaction between people in social events, starting from gestures, body movements and speech [14].

1.1 Form-based methods

Human body parts can be described in different ways in 2D-space and in 3D-space as rectangular areas, sets of coordinates of certain points and articulations, as volumetric figures (see Fig. 3). It is well known that activity recognition algorithms based on human silhouette (shape-based) are becoming more and more popular with the advent of neural networks, but due to the use of data of only one modality, it is necessary to recognize human body parts with high accuracy.

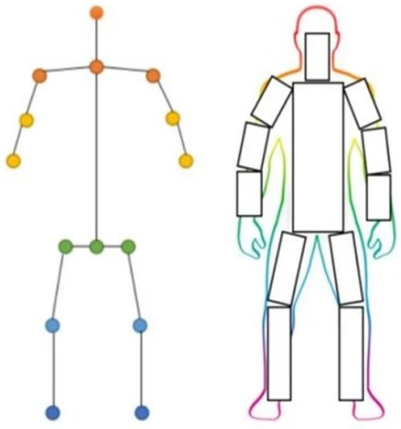
Figure 3. Human representation in 2D and 3D space

One of the key features of this technique, based on skeletal structure extraction, is the representation of the human body and its skeleton by a set of coordinates in space, which is a de-

parture from traditional pixel-based representations. This approach not only reduces the dimensionality of the original data, but also captures the essential spatial and temporal features needed for action recognition.

Actions can be classified in quite a few different ways, but the most common methods are: 1) frame voting method, 2) global histogram method, 3) SVM classification method and 4) dynamic time warping method.

Graphical models are widely used in 3D modeling of human pose. Combining discriminative and generative models improves human pose estimation.

Recognition procedures can be implemented in real time using stepwise covariance updating and nearest neighbor classification methods. Human pose estimation is very sensitive to a variety of circumstances including changes in lighting, viewpoints, occlusions, background clutter, and human clothing. Low-cost technologies such as Microsoft Kinect and other RGB-D sensors can effectively combat these limitations and provide reasonably accurate estimation.

1.2 Structure of the recognition system based on skeletal representation

Based on the technological capabilities and efficiency of the considered methods of human activity categorization, it was decided to focus on the approach using the skeletal structure representation. Effective implementation of the action recognition system involves analysis of the video stream with subsequent extraction of feature data. Figure 4 shows the authors' proposed structure of the recognition system.

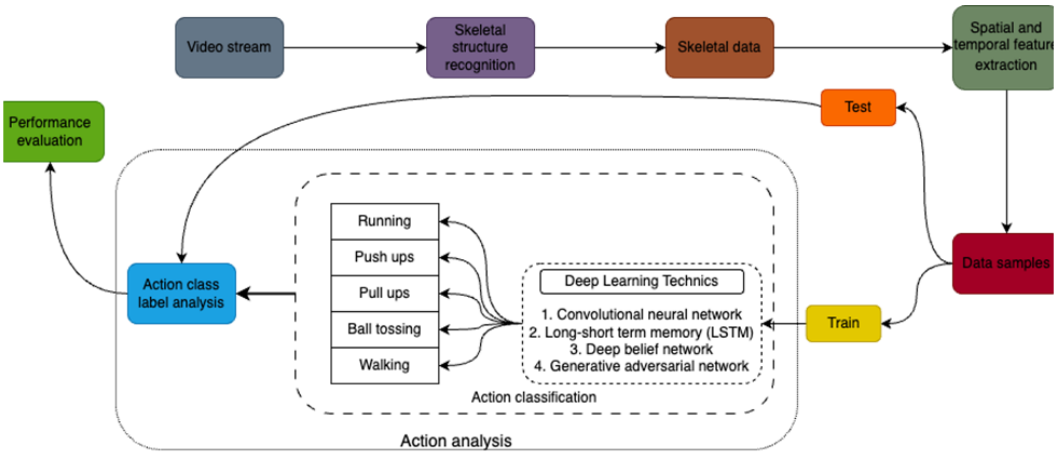

Figure 4. Action recognition system concept

The proposed system architecture consists of three main stages: 1) skeleton extraction, 2) spatial and temporal feature extraction, and 3) activity recognition.

It should be noted, however, that signs are extracted by tracing key joints in the human body.

2. Skeletal structure feature extraction platforms

With the development of artificial intelligence technologies, a large number of pre-trained models and platforms have emerged that enable fast and high-quality execution of commonly encountered machine learning subtasks. One such subtask is analyzing an image or video stream and identifying the skeletal structure of a person.

Currently, there are a sufficient number of platforms available in the public domain for use that solve the coordinate extraction problem. These include platforms such as OpenPose [15], Detectron2 [16], MediaPipe [17] and YOLOv7 [18]. The most popular tools are OpenPose, MediaPipe and YOLOv7 Pose.

MediaPipe is an open-source system from Google for building cross-platform customizable machine learning solutions for live and streaming media. MediaPipe is currently under active development and contains extensive documentation, including demonstrations and examples of how to use the built-in features. The system uses the BlazePose 33 benchmark topology. BlazePose is a set of 3 topologies: the COCO keypoints, Blaze Palm, and Blaze Face. It works in two stages: detection and tracking. Since detection is not done in every frame, MediaPipe can perform the output faster. MediaPipe uses three models for pose estimation.

OpenPose is an open-source real-time multi-person detection system for collaborative detection of key points on the human body, palms, face and feet. This project relies heavily on the CMU Panoptic Studio dataset. OpenPose also includes demonstrations and examples of how to use the built-in features.

The YOLO (You Only Look Once) v7 is the latest in the YOLO family of models. YOLO models are single-stage object detectors. In YOLO, image frames are represented through feature extraction. These features are combined and blended and then passed to the head of the network. This model predicts the locations and classes of objects around which bounding boxes should be drawn.

2.1 Comparative analysis of the platforms

The authors of this paper compare the platforms in the context of applicability for analyzing video of sports exercises performed by a single athlete in a frame.

According to a study by Radzki P. [19], all solutions have good accuracy in detecting the articulations of the human body when rendering relatively static images; in poor lighting or regardless of whether the person is looking directly at the camera.

The biggest problem in the skeletal structure recognition task is motion blur, which, along with increasing motion speed, leads to large errors in the representation of landmark positions, up to a complete loss of detection. In this area, MediaPipe proved to be much more effective in dealing with failures. In this test, MediaPipe showed significantly more robustness to blurring than OpenPose. Figure 5 shows the effect of motion blur on landmark detection.

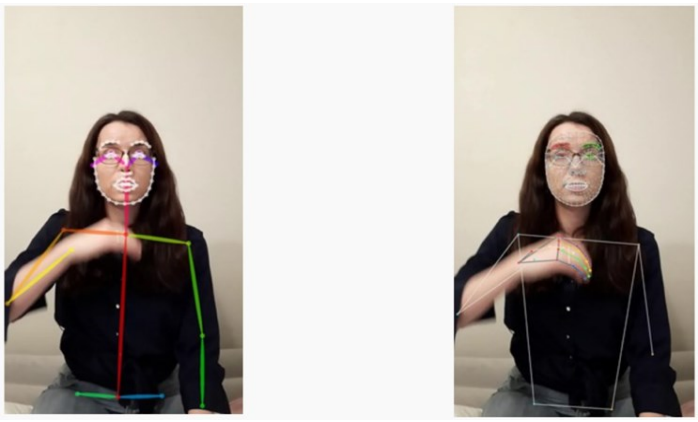

Figure 5. Example of OpenPose and MediPipe operation at blurring

Kukil and V. Gupta [20] note that YOLOv7 Pose performs worse at detecting a human figure on a small scale compared to MediaPipe. In Figure 6, it is noticeable that the model was not found to be able to detect a human being in all frames in the video stream losing sight of it when the size of the human figure in the frame is reduced. MediaPipe, in turn, was able to detect the person at a much smaller scale.

Figure 6. Example of YOLOv7 and MediaPipe working with small scale

This problem is caused by the difference in the pose estimation methods used by the frameworks. MediaPipe tracks a person after confirming object detection using an auxiliary network. On the other hand, YOLOv7 performs detection for each frame separately, which reduces its quality but significantly increases the computational speed.

All models have their advantages and disadvantages. The OpenPose model requires large computing power and special platforms to operate, but its accuracy is high due to the ability to determine the pose of a person in each frame of the network. In addition, compared to the MediaPipe model, its training takes a long time. The OpenPose model only outputs 2D points, so it cannot accurately detect the angles of some joints, so OpenPose is suitable for high-precision tasks where real-time tracking is neglected. On the other hand, the MediaPipe model is lightweight and fast and does not require installation on any platforms. Although the accuracy is lower than OpenPose, training and deployment takes less time: using an auxiliary detector, the pipeline locates the Region of Interest (in this case, a human figure) (ROI) in the frame, then the tracker predicts the pose landmarks and segmentation mask within the ROI, using as input an image cropped over the region of interest, thus not considering redundant information. MediaPipe outputs 3D points, so it accurately determines the angles between joints and displays them on the screen. For this reason, MediaPipe is suitable for real-time applications. Compared to YOLOv7, MediaPipe performs well on low resolution input data. MediaPipe is faster than YOLOv7 in handling CPU pins, it also performs relatively well in detecting remote objects (in our case, people). However, when it comes to occlusion, YOLOv7 performs better. YOLOv7 also performs better in analyzing fast movements, in the case of high-resolution input images.

YOLOv7 and OpenPose are frameworks for multi-person detection. MediaPipe does not have this functionality, but for the task of recognizing one single athlete in a frame, this limitation is not a problem.

Based on the analyzed characteristics of each platform, it was concluded that the most suitable platform for modeling the skeletal structure of a single athlete from a video stream is MediaPipe due to its representation of the skeletal structure in 3D space, which allows more accurate calculation of angles between skeletal points and evaluation of the quality of exercise performance.

2.2 Extractable skeletal structure features with MediPipe

The landmark model in MediaPipe Pose predicts the location of 33 pose landmarks. The points to be detected are shown in Figure 7

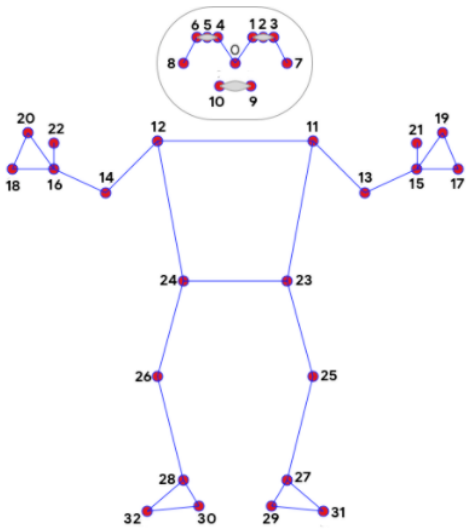

Figure 7. Recognizable points of the skeletal structure

The model returns the following data for each landmark:

- x and y: landmark coordinates normalized [0.0, 1.0] by image width and height, respectively.
- z: represents the depth of the landmark with the depth at mid-thigh as the starting point, and the smaller the value, the closer the landmark is to the camera. The z value uses approximately the same scale as x.
- visibility: a value [0.0, 1.0] indicating the probability that the landmark is visible (present and not overlapped) in the image.

The video was processed using Python programming language and OpenCV [21], Numpy [22], MediaPipe libraries.

3. Proposed methodology for recognizing athlete's actions on video

The authors propose a methodology based on a deep learning framework based on layers with long-term memory LSTM [23].

LSTM is a specialized recurrent neural network layer that was specifically designed to replace the conventional recurrent layer for working with sequential data sets where long-term temporal dependencies between individual data packets are important. Examples of such data could be text, audio, image (video) sequences or time series.

This neural network architecture allows an input packet of information to be processed based on "knowledge" of previous processed packets. The LSTM module deepens this concept and introduces new parts of the module, referred to as "cell state", which can retain information for long periods of time. This cell state is controlled by three memory gates: input gate, forget gate, and output gate. These "gates" determine what information should be retained or discarded from the cell state.

The input gate is responsible for determining the information to be added to the cell state, while the forget gate is responsible for removing pieces of information from the cell state. The output gate decides what information to remove from the cell state.

The LSTM architecture was chosen for its ability to capture temporal dependencies in sequential data, making it particularly suitable for the dynamic nature of human movements. This neural network is optimized for both accuracy and interpretability, which is consistent with the overall goals of improving the understanding of complex human activity.

The training dataset consists of sets of videos corresponding to one of the 15 movement categories selected as examples for testing the approach. Within a video there is only one person performing the actions. The dataset UCF101 - Action Recognition Data Set [24], which is publicly available, was used as a basis.

The video processing algorithm was as follows:

1) Iterate over all available videos frame by frame;
2) Apply the human skeletal structure recognition model;
3) Extract articulation coordinates;
4) Add to the frame information about what action class it belongs to (1 of 15);
5) Save the extracted information to a file with a special extension ".npy".

3.1 Dataset collection and processing

UCF101 is a dataset designed specifically for the task of human action recognition, consisting of realistic short videos collected from the YouTube platform with 101 categories of different actions from different areas of human activity. This dataset contains 13,320 videos that provide the greatest diversity in terms of actions, camera motion variations, object appearance and pose, object scale, viewpoint, cluttered background, lighting conditions, etc.

Videos in the dataset are grouped into small clusters of 4-7 pieces. Videos from the same group may have some common features, such as background, viewing angle, lighting, distance of objects, etc.

In this paper, processing was conducted on 15 classes: 1) Archery, 2) BenchPress, 3) Biking, 4) BodyWeightSquats, 5) BoxingPunchingBag, 6) Drumming, 7) HandstandPushups, 8) HandstandWalking, 9) HighJump, 10) HorseRiding, 11) JavelinThrow, 12) JumpingJack, 13) PullUps, 14) PushUps, 15) WalkingWithDog.

For each video, a separate ".npy" file with an appropriate name is collected, which stores line-by-line information about x, y, z coordinates of the skeletal structure recognized in each individual frame.

3.2 LSTM neural network for classification

The analysis of body motion over time and prediction is done using the LSTM network. So, key points from a sequence of frames are sent to LSTM for motion classification. The LSTM model, which is used to classify actions based on key points, is trained using the Tensorflow machine learning library [25].

The input data for training contains a sequence of keypoints (33 keypoints per frame) and the corresponding action labels. A continuous sequence of 24 frames is used to identify a particular action. An example sequence of 24 frames would be a multidimensional array of size 24×99 as follows:

$$[[x_0 y_0 z_0 \dots\ x_{98} y_{98}\ z_{98}]\ [x_0 \dots [x_0 y_0 \dots y_0 z_0 \dots z_0 \dots\ \dots\ \dots x_{98} \dots x_{98} y_{98} \dots y_{98} z_{98}] \dots z_{98}]] \quad (1)$$

Each row contains 33 key point values. Each key point is represented by (x, y, z) values, hence a total of 99 values in a row.

For training, the neural network was configured as follows:

Table 1 – neural network layer sequence

Layer	Output dimension	Number of parameters
LSTM	(24, 128)	116736
Dropout	(24, 128)	0
LSTM	(24, 256)	394240
Dropout	(24, 256)	0
LSTM	256	525312
Batch_normalization	256	1024
Dense	256	65792
Dense	128	32896
Dense	64	8256
Dense	15	975

Total number of parameters: 1 145 231

Training number of parameters: 1 144 719

Non-learning parameters: 512

Adam optimizer, error function - categorical cross - entropy loss and categorical accuracy metric were used during training. According to Kingma D. P. and B. J. Adam [26], this method is "computationally efficient, requires little memory, is invariant to diagonal scaling of gradients, and is well suited for data/parameter large problems".

Categorical cross-entropy is used in this paper as a loss function for a multi-class classification model with two or more output labels [27]. The output label is assigned a coding value of one of the categories in the form of 0 or 1. The output label, if present in integer form, is converted to categorical coding using the method.

Categorical accuracy considers the frequency of matching predictions to real labels.

The model was trained for 56 epochs and trained for about an hour.

The model training stages were observed through the TensorBoard visualization tool [28]. TensorBoard is a set of web-based applications for validating and understanding your TensorFlow model runs and plots. TensorBoard is designed to work completely offline, requiring no Internet access. TensorBoard provides the user with the necessary functionality to visualize their experiments, as well as the tools needed to experiment with machine learning:

- Tracking and visualization of quality and error metrics as a function of algorithm iteration/age;
- Visualization of the dynamic/static model graph (operations and layers);
- Media data display (images, text and audio data);
- Profiling of TensorFlow programs.

During model training, we plotted the error function versus execution epoch of the algorithm on the training data (red) and the error function plot on the validation data (blue) (see Figure 8).

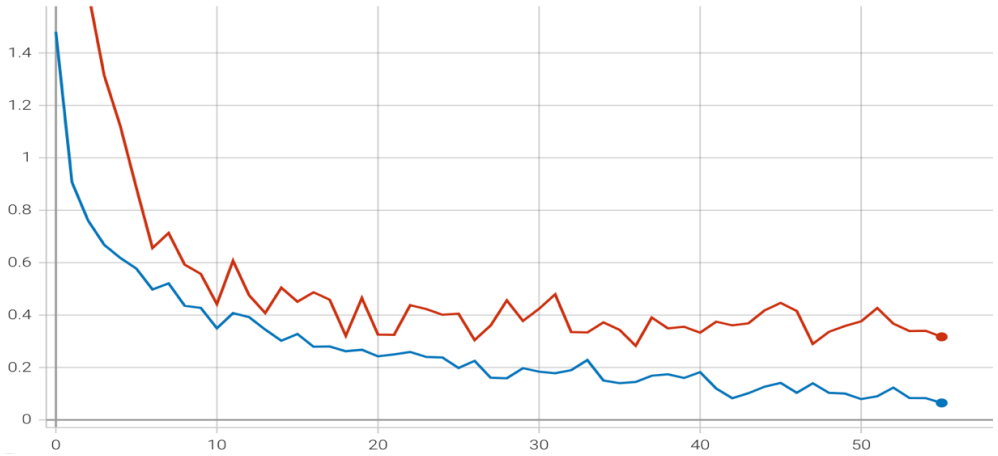

Figure 8. Graph of dependence of the error function on the algorithm execution epoch

The construction of two such graphs allow us to find the optimal point of stopping training, i.e. the moment when further training of the machine learning model ceases to bring significant improvement of quality metrics (calculated on the selected and suitable for the specific task) on the test dataset, and retraining on the training dataset begins. One indication of model overtraining is a decrease in error on the training set and an increase/return to plateau on. The point where the error on the test dataset begins to increase is generally considered the optimal place to stop training.

We also plotted graphs showing changes in prediction quality over the course of the training iteration, by categorical accuracy metric (see Figure 9).

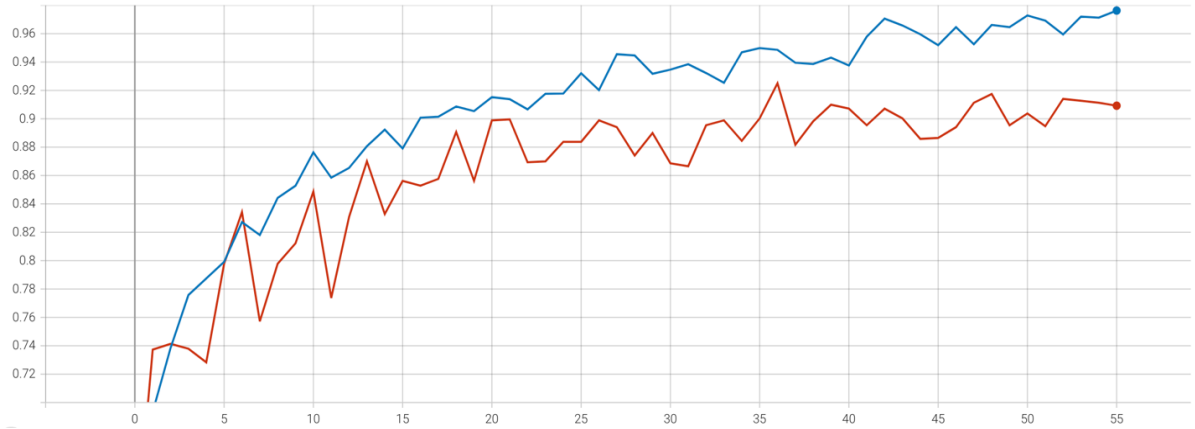

Figure 9. Graph of dependence of prediction accuracy on the algorithm execution epoch

3.3 Training results

While training the network, the metrics precision (2), recall (3) and the loss function for training, testing and validation are calculated to get an idea of how well the network is trained. Finally, to validate the performance of the model, about 1000 test videos are fed on which the metrics are calculated, at the end an error matrix is presented to visualize the recognition quality.

$$precision(a, X) = \frac{TP}{TP + FP} \tag{2}$$

$$recall(a, X) = \frac{TP}{TP + FN} \tag{3}$$

Results:

- Training loss: 0.0647
- Training categorical accuracy: 0.9763
- Validation loss: 0.3169
- Validation categorical accuracy: 0.9092

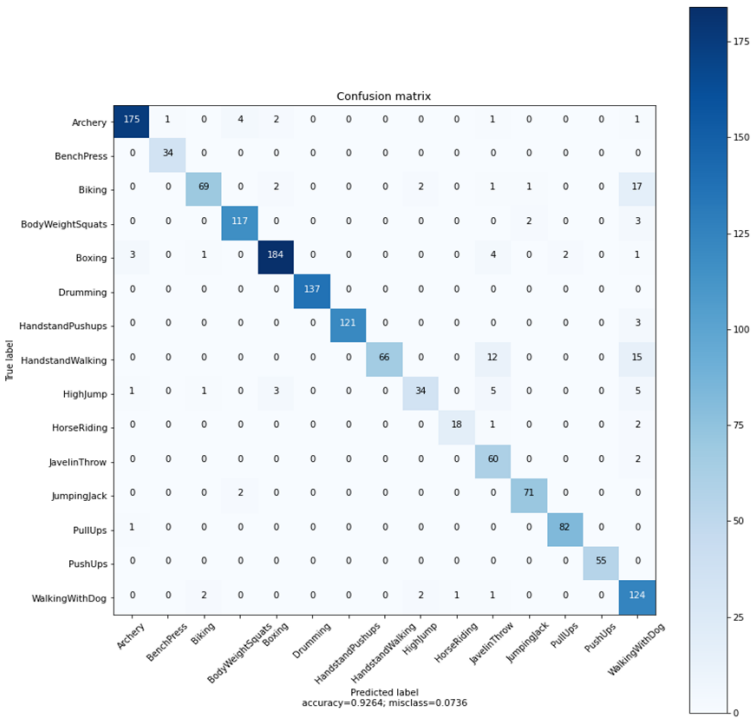

Figure 10. Confusion matrix

Together with the construction of the error matrix (see Figure 10), accuracy and completeness allow us to interpret the results well, to trace the correlation between the algorithm's predictions and the true answer.

Figure 11 shows the results on videos belonging to different classes of actions.

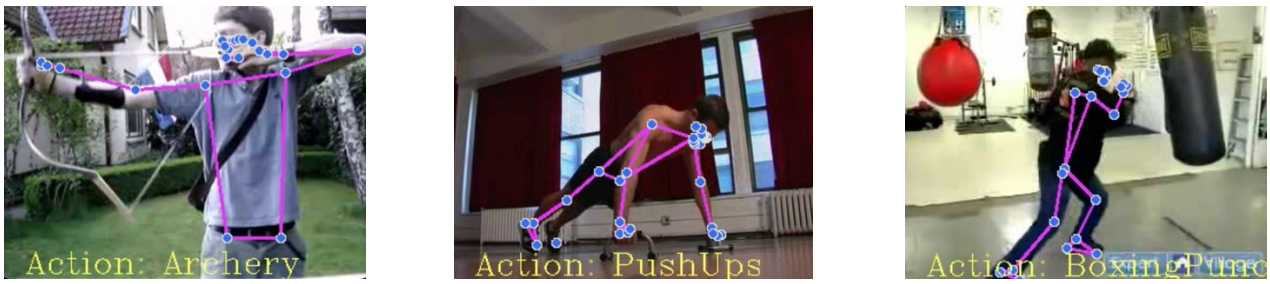

Figure 11. Example of model operation by video stream on different classes

4. Conclusion

This paper analyzed methodologies for human motion recognition in video. According to many researchers, one of the most promising is the shape-based methodology. As a result of the comparative analysis of skeletal structure extraction platforms, MediaPipe is the most suitable platform for the task of recognizing sports exercises performed by the only person in the frame. It has advantages in CPU speed, processing of artifacts in the video and gives 3D point coordinates, which allows for more accurate calculation of angles between skeletal articulations, thereby allowing for a more accurate assessment of the quality of the athlete's exercise performance.

The authors of the paper propose their own implementation of an approach based on the analysis of human body shape and location in space, using the skeletal structure extraction platform MediaPipe and a neural network based on layers with long-term memory LSTM. This implementation shows sufficient quality of real-time action recognition on UCF101 dataset. This methodology can be applied for modeling athletes' poses in 3D space, visualizing their skeletal structure, which will allow to interpret results and monitor the quality of the performed exercises in automatic mode without the presence of a coach.

Other applications of human motion recognition technology include use in the medical field to analyze patient movements for rehabilitation and diagnose patient positions, in the gaming industry to create virtual characters, track player movements, enhance and create a unique gaming experience, and in the security field to monitor and detect suspicious activity on video by embedding this technology in surveillance cameras.

With the ability to visualize skeletal structure, researchers and experts can more clearly represent and analyze motion data, which greatly simplifies the process of interpreting results and increases their comprehensibility, as well as simplifies user interaction with any systems in which this technology will be applied, due to the simpler visual perception.

5. Acknowledgements

The work was financially supported by RSF (grant № 22-11-00213, https://rscf.ru/project/22-11-00213/).

References:

1. Laith A., Jinglan Zhang. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data. №8 (53). 2021.

2. M. Vrigkas, C. Nikou, I. A. Kakadiaris. A Review of Human Activity Recognition Methods. Robot. AI. Vol. 2. 2015.

3. Gavrila, D. M. The Visual Analysis of Human Movement: A Survey // Computer Vision and Image Understanding. 1999. №1. P. 82–98.

4. Aggarwal J.K., Cai Q. Human Motion Analysis: A Review // Computer Vision and Image Understanding. 1999. №3. P. 428–440.
5. Aggarwal J.K., Xia L. Human activity recognition from 3D data: A review // Pattern Recognition Letters. 2014. №48. P. 70–80.
6. Ghotkar A., Vidap P., Deo K. Dynamic Hand Gesture Recognition using Hidden Markov Model by Microsoft Kinect Sensor // International Journal of Computer Applications. 2016. №5. P. 5–9.
7. Li R., Zickler T. Discriminative virtual views for cross-view action recognition. IEEE Conference on Computer Vision and Pattern Recognition. 2012. P. 2855–2862.
8. Iosifidis A., Tefas A., Pitas I. Activity-based person identification using fuzzy representation and discriminant learning // EEE Transactions on Information Forensics and Security. 2012. Vol. 7. №2. P. 530–542.
9. Chen C. Y. Grauman, K. Efficient activity detection with max-subgraph search. EEE Conference on Computer Vision and Pattern Recognition. 2012. P. 1274–1281.
10. Tran K. N., Kakadiaris I. A., Shah S. K. Part-based motion descriptor image for human action recognition // Pattern Recognition. 2012. №7. P. 2562–2572.
11. Wu Q., Wang Z., Deng F., Chi Z., Feng D. D. Realistic human action recognition with multimodal feature selection and fusion // EEE Transactions on Systems, Man, and Cybernetics: Systems. 2013. №4. P. 875–885.
12. Martinez H. P., Yannakakis G. N., Hallam J. (2014). Don't Classify Ratings of Affect; Rank Them! // IEEE Transactions on Affective Computing. 2014. № 5. P. 314–326.
13. Vrigkas M., Nikou C., Kakadiaris I. A. Classifying Behavioral Attributes Using Conditional Random Fields // Artificial Intelligence: Methods and Applications. 2014. №8445. P. 95–104.
14. Marín-Jiménez M. J., etc. Human interaction categorization by using audio-visual cues // Machine Vision and Applications volume. 2014. №25. P. 71–84.
15. OpenPose Documentation. OpenPose. 2022. URL: https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/index.html (accessed: 01.12.2022).
16. Detectron2. Meta AI. 2022. URL: https://ai.facebook.com/tools/detectron2/ (accessed: 03.12.2022).
17. MediaPipe Pose. MediaPipe. 2022. URL: https://google.github.io/mediapipe/solutions/pose.html (accessed: 05.12.2022).
18. Wang C.Y., Bochkovskiy A., Liao H. Y. YOLOv7: Trainable Bag-Of-Freebies Sets New State-Of-The-Art for Real-Time Object Detectors // Computer Vision and Pattern Recognition. 2022. 15 p. (https://arxiv.org/abs/2207.02696)
19. Radzki P. Detection of Human Body Landmarks - Mediapipe and Openpose Comparison // HearAI. 03.04.2022. URL: https://www.hearai.pl/post/14-openpose/ (accessed: 05.12.2022).
20. Kukil, V. Gupta. YOLOv7 Pose vs MediaPipe in Human Pose Estimation. LearnOpenCV. 2022. URL: https://learnopencv.com/yolov7-pose-vs-mediapipe-in-human-pose-estimation/ (accessed: 11.10.2022).
21. OpenCV. Intel. 2021. URL: https://opencv.org/ (accessed: 11.10.2022).
22. NumPy Documentation. NumPy. 2022. URL: https://numpy.org/doc/stable/ (accessed: 23.10.2022).
23. Shipra S. What is LSTM? Introduction to Long Short-Term Memory // Analytics Vidhya. URL: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/ (accessed: 16.03.2024)
24. UCF101 - Action Recognition Data Set. UCF: Center for Research in Computer Vision. 2013. URL: https://www.crcv.ucf.edu/data/UCF101.php (accessed: 22.10.2022).
25. TensorFlow Documentation. TensorFlow. 2022. URL: https://www.tensorflow.org/api_docs (accessed: 10.12.2022).
26. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization // Machine Learning. Conference Paper at ICLR. 2015. 15 p. (https://arxiv.org/abs/1412.6980)

27. Kumar A. Keras – Categorical Cross Entropy Loss Function // Data Analytics. 28.10.2020. URL: https://vitalflux.com/keras-categorical-cross-entropy-loss-function/ (accessed: 15.12.2022).
28. TensorBoard. TensorFlow. 2022. URL: https://www.tensorflow.org/tensorboard (accessed: 17.12.2022).