# Understanding the Motion Adaption of Machine Using Long Short – Term Memory Networks for voiceless Virtual Assistant.

**Subhrajit Roy[1], Binoy Das[2]**

1 Department of CSE, NIELIT
Agartala, Tripura, India,
Subhrajitroy13@gmail.com

2 Department of CSE, NIELIT
Agartala, Tripura, India,
erbinoy@nielit.gov.in

*Abstract:*

*In recent years, the study of computer vision and pattern recognition has seen a significant increase in the popularity of video-based human action recognition as a research topic. Many different fields, including surveillance, robotics, healthcare, video searching, and human-computer interaction, are among its many potential uses. Human action identification in videos faces several difficulties, including crowded backdrops, occlusions, viewpoint fluctuation, execution rate, and camera motion. Over the years, numerous strategies have been put up to deal with the difficulties. For research, three different dataset types—single perspective, multiple viewpoints, and RGB-depth videos—are used. This paper provides an overview of several cutting-edge deep learning-based methods for the recognition of human actions on three different kinds of datasets. Given the increasing.[6]. Here we are using Long short-term Memory networks, as it is a part of Neural Networks. It is more efficient and accurate to create the structures so that the model can understand the method easily.*

*Keywords: Holistic, OpenCV, LSTM, RNN, Numpy, mediapipe, os.*

## I. INTRODUCTION

The difficult task of identifying human actions in a video stream has recently attracted a lot of interest from the computer vision research community. Analysing a human action involves more than just demonstrating how various bodily parts move;[1][2] it also involves describing the aim, emotion, and thoughts of the actor. As a result, it is now an important part of understanding and analysing human behaviour, which is important in many fields, such as surveillance, robotics, healthcare, video searching, and human-computer interaction.[4] Video data differs from static images in that it incorporates temporal information, which is crucial for recognising actions.[3][5]Video data also incorporates naturally occurring data augmentation, such as jittering for video frame classification.

Lately, much work has been finished in various regions in the PC vision research region, like video Arrangement, goal and division and so forth. In any case, the examination on video-based human movement acknowledgment has not been investigated a lot, because of the difficulties in handling fleeting data from the video transfer.[6][11] Activity acknowledgment from a video transfer can characterized as perceive human activities naturally utilizing an example acknowledgment framework with insignificant human-PC cooperation.[10][11] Commonly, an activity acknowledgment framework dissects specific video groupings or edges to get familiar with the examples of a specific human activity in the preparing interaction and utilize the learnt information to order comparable activities during the testing stage.

## II. RELATED WORK

### A. Motion features comparison

In the majority of comparable work as well as in this work, the spatiotemporal motion data is standardised to become independent of the subject's position, orientation, and skeleton size. [33] As shown in fig. 1, the root joint is pinned into the origin (0, 0, 0) to rotate the skeleton in each frame so that its hips face the fixed orientation. The normalised data is subjected to additional processing [10, 31, 41, 46, 47] to produce a descriptive frame-based.

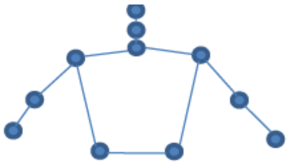


Fig 1: Upper body structure with 11 joints

### B. Annotation of Segments

The process of gradually dividing the videocapture() functions into segments , is known as segmentation. [4, 11, 31] The classic approach of a mediapipe() function that is shifted systematically and rewritten is used to identify the segments, resulting in a video capture as reproduction datasets. [11] A number of simultaneous mediapipe() that are various heights and widths enhance data replication by 20 times. A nonlinear segmentation is suggested to produce segments of various lengths with semantic power in order to prevent a significant segmentation[6] overlap. In order to acquire the closest preset data samples, the segments must be processed in order to extract segment level features. The nearest-match class is taken into consideration if there is a high degree of similarity between a segment and its closest match. [11, 31] It can be computationally taxing to compare the features of several segments with numerous action samples, especially when using the pricey read and holistic function of quadratic time program complexity.

### C. Detection and Prediction

Movements of hand and upperbody can be annotated in live camera using machine learning based approaches , [46] which means that the average frame sampling degree is higher than the actual frame sampling degree rate. The very basic need is that Machine Learning based sample collection techniques [7], are also suited for early detection that is, [41] finding the beginnings of activities even before they are completed. Movements can even be observed many times in advance by raising acceptance sensitivity, but this usually comes at the expense of decreased recognition accuracy. In time-sensitive applications, such as gesture detection for gaming, action prediction is crucial.

### D. Our contributions

We use deep learning(LSTM neural networks), which was previously demonstrated efficacy on identifying brief portioned movements , to dramatically outperform the existing methods for annotating skeletal sequences. [24, 27, 29] Our online action identification method (Online-LSTM), [24, 37] in particular, can accurately detect the starting and ending of serial activities inside the MPHolostic(). In this research paper we have demonstrated that actions may be recognised at their beginnings without having to wait for them to complete, to boost the accuracy of acceptable values, activities may be anticipated a 100ms in advance.

The findings are evaluated using the metrics contrasted by using cutting-edge methods. Our method is more than one degree efficient than the existing cutting-edge motion detection algorithms, achieving not only a distinctly higher effectiveness but also demonstrating superior performance.

## III. METHODOLOGY

### A. Problem Defination

A set of consecutive skeleton postures (P1...... Pn) is used to illustrate a motion sequence (or just motion) Pi (i 1, n). The motion length is determined by the total number of skeletal poses, n. The 31 monitored joints of the i$^{th}$ position with the P, which was instant (Pi € R93) are shown visually in acquired at time I (1 I n), which are shown visually in Fig. 1 as the 3D coordinates of 31 tracked joints. We will track the movements so that will help us to create a concreate dataset which will be more efficient to train the model to read the actions of a human. These actions are not general they are the signs and symbols used by deaf and verbally nonprivileged persons and later worked as their very own personal assistant. [28, 32, 47].

A pseudo-infinite motion sequence is known as a stream Only very small number of previous frames may be read in main memory at any given time, and also these are not processed.

Neural networks are the most efficient way to resolve structure communication and confusion matrix problems. Face recognition is one of them and if we want to resolve the problem, we need to implement the neural network problems. Neural network has many forms among them we have used Long Short- Term because this algorithm provides more accurate values for human movement with graph. We have discussed about the algorithm below.

## IV. EVALUATION OF THE PROCESS AND EXPARIMENT

Experimental analysis of the effectiveness and efficiency of the suggested Online-LSTM and Offline-LSTM models' annotation quality. These Machine Learning models are assessed using 3 separate real world applications: offline sequence action Detection using segments, action early-detection and prediction, and real-time stream annotation. Additionally, 3 different datasets which contains subsets also (created Live and saved) are used to evaluate each use case [30,46,47]. This dataset is being used because it identifies total 270 number of classes, that are the highest number of classes compared with other datasets, and because it gives ground reality of the algorithm that is suited for the evaluation of Machine Learning algorithms. The best outcomes are contrasted with cutting-edge methods assessed on the identical dataset.

### *A. Dataset*

The Dataset that we have created by tracking the skeleton movements and we have saved the dataset on a sequence of 30x30 matrix. All these values are predefined by the different human and machine interactions, we have used different persons to provide different signs of especially abled people and helped us to create an efficient system for them. [42,45,48]

TABLE1. DESCRIPTIONS OF DATASET

| **Model: "sequential_2"** | | |
|---|---|---|
| ***Layer _Segments*** | ***Output stream*** | ***Parameter*** |
| lstm6 (LSTM) | (None, 30, 64) | 442112 |
| lstm7 (LSTM) | (None, 30, 128) | 98816 |
| lstm8 (LSTM) | (None, 64) | 49408 |
| dense6 (Dense) | (None, 64) | 4160 |
| dense7 (Dense) | (None, 32) | 2080 |
| dense8 (Dense) | (None, 3) | 99 |

We have collected total samples of: 596,675 and Trainable params: 596,675 and Non-trainable params: 0

### *B. Proposed Algorithm:*

The proposed algorithm is based on LSTM and here we need to find the graphs for both body, hands, face[34,35,36] and it is also calculating the body poses. Hence, we need to create dataset for the machine so that the machine can easily understand the predictions on the basis of the dataset we have created. [38,39] The numpy array is used to create one of the major matrices to understand the mathematics behind the confusion matrix.to train the model we have used the real time data so that the predictions could more detailed and accurate.  In details the algorithm has been described below-

```
1.  Collect_Feathers <- FaceMesh_Feathers or Face Feathers
2.  Collect_Pose <- Pose_Connections or body Pose
3.  Collect_Left_hand_Connections <- Hand_Connections
4.  Collect_Right_hand_Connections <- Hand_Connections
5.  Declare two variables to store results.
6.  Use Videocapture function ()
7.  Draw land Marks.
8.  Draw_Landmarks <- (frame, results)
9.  Extract Key point values for
    a.  Body <- np. Array
    b.  Face<- np. Array
    c.  Left_hand<- np. Array
    d.  Right_hand<- np. Array
10. Setup folders for data collection.
11. Pre-process the data and create labels and feathers.
12. Build and train the LSTM network(Tensorflow)
    a.  Import<- Sequential
    b.  Import<- LSTM
    c.  Import<-Dense
    d.  Import<-Tensorboard
13. Make Predections
    a.  Res <- Model. Predection
```

b. Save waights<- model.save

14. Evaluation using confusion Matrix and Accuracy
15. Test performed in real time.

*C. Protocols used for experiments*

First portion of the algorithm is used to train the model and the second part of the algorithm is used to test the annotation in the first part it detects the only upper body and hands and second part it detects the facial graphs so the expression of the user also can be detected very precisely.

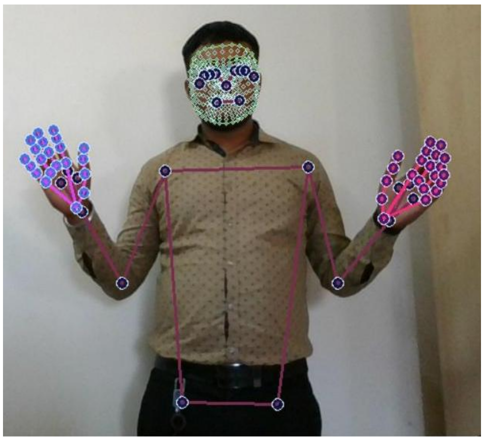


Fig 2. Providing Training to the model using Skeleton graph

*D. Test Results Representation using confussion matrix:*

array([ 0.46401805, 0.47148561, -0.42863482, ..., 0.33942375,
0.26005563, -0.00118776])

And the Numpy Array is

array([ 0.46401805, 0.47148561, -0.42863482, ..., 0.33942375,
0.26005563, -0.00118776])

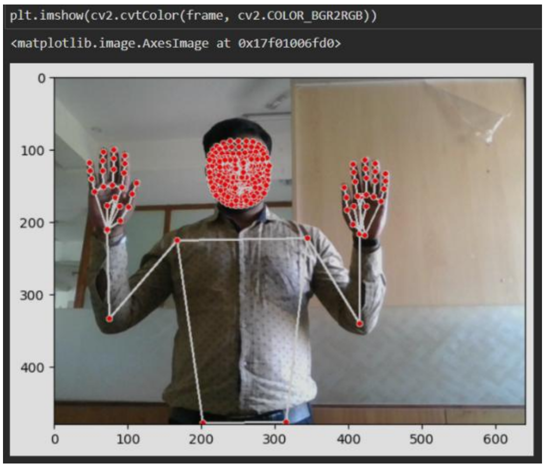
*E. Label Map of the model:*

{'hello': 0, 'thanks': 1, 'iloveyou': 2}

Shape of the numpy array of the image :

(180, 30, 1662)

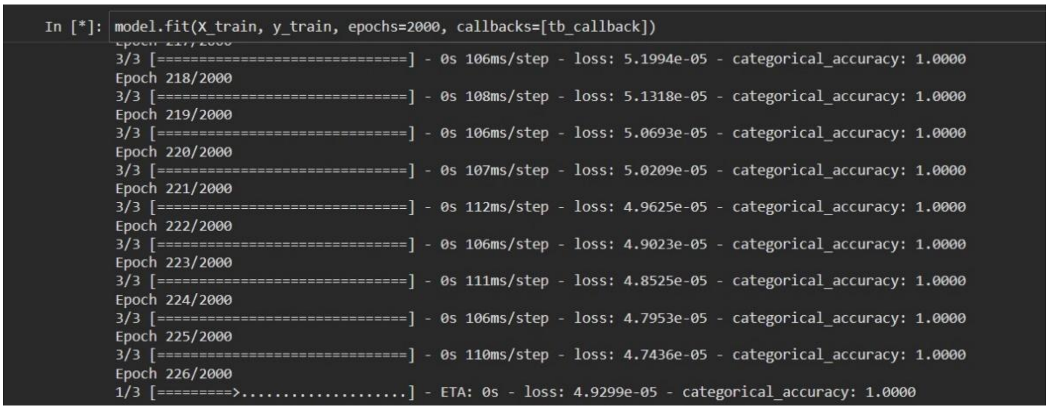
Simply said, training a model entails learning (deciding) appropriate values for each weight and bias from labelled samples. [24,25] Empirical risk minimization is the technique by which a machine learning algorithm constructs a model in supervised learning by analysing several examples and looking for a model that minimises loss.



```
plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x17f01006fd0>
```

Fig 3. Mathematical analysis of the model using confussion matrix.

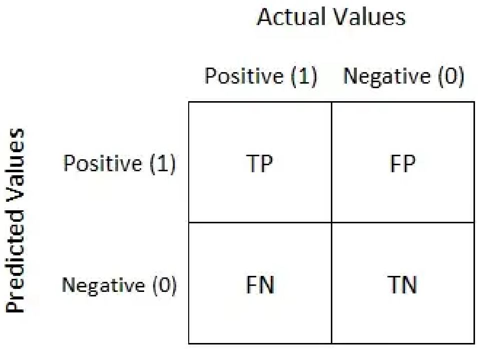


```
In [*]: model.fit(X_train, y_train, epochs=2000, callbacks=[tb_callback])
        3/3 [==============================] - 0s 106ms/step - loss: 5.1994e-05 - categorical_accuracy: 1.0000
        Epoch 218/2000
        3/3 [==============================] - 0s 108ms/step - loss: 5.1318e-05 - categorical_accuracy: 1.0000
        Epoch 219/2000
        3/3 [==============================] - 0s 106ms/step - loss: 5.0693e-05 - categorical_accuracy: 1.0000
        Epoch 220/2000
        3/3 [==============================] - 0s 107ms/step - loss: 5.0209e-05 - categorical_accuracy: 1.0000
        Epoch 221/2000
        3/3 [==============================] - 0s 112ms/step - loss: 4.9625e-05 - categorical_accuracy: 1.0000
        Epoch 222/2000
        3/3 [==============================] - 0s 106ms/step - loss: 4.9023e-05 - categorical_accuracy: 1.0000
        Epoch 223/2000
        3/3 [==============================] - 0s 111ms/step - loss: 4.8525e-05 - categorical_accuracy: 1.0000
        Epoch 224/2000
        3/3 [==============================] - 0s 106ms/step - loss: 4.7953e-05 - categorical_accuracy: 1.0000
        Epoch 225/2000
        3/3 [==============================] - 0s 110ms/step - loss: 4.7436e-05 - categorical_accuracy: 1.0000
        Epoch 226/2000
        1/3 [=========>....................] - ETA: 0s - loss: 4.9299e-05 - categorical_accuracy: 1.0000
```

Fig 4. Training the model using the dataset that is created .

## V. CONFUSION MATRIX

After cleaning, pre-processing, and wrangling the data, the first thing we do is feed it to a fantastic model, [26,27,28] which, of course, produces results in probabilities. How in the world are we going to evaluate the performance of our model. That is exactly what we want: greater efficacy and performance. [29,30] And this is where the Confusion Matrix is highlighted. A machine learning classification performance metric is the confusion matrix.



| | | Actual Values | |
|---|---|---|---|
| | | Positive (1) | Negative (0) |
| Predicted Values | Positive (1) | TP | FP |
| | Negative (0) | FN | TN |

Fig 5. Representation of Confussiopn Matrix

### A. Results

We have provided the results with the matrix representation along side the real time test.[11,12] We have focused to the novel cause and that gives us continuous motivation to create a model for specially abled persons so that they can be also feels like the part of the technological advancements . [15]

i. Mathematical Representation:

```
array([[[2, 0],
        [0, 3]],
       [[4, 0],
        [0, 1]],
       [[4, 0],
        [0, 1]]], dtype=int64)
```

ii. Pictorial Representation:

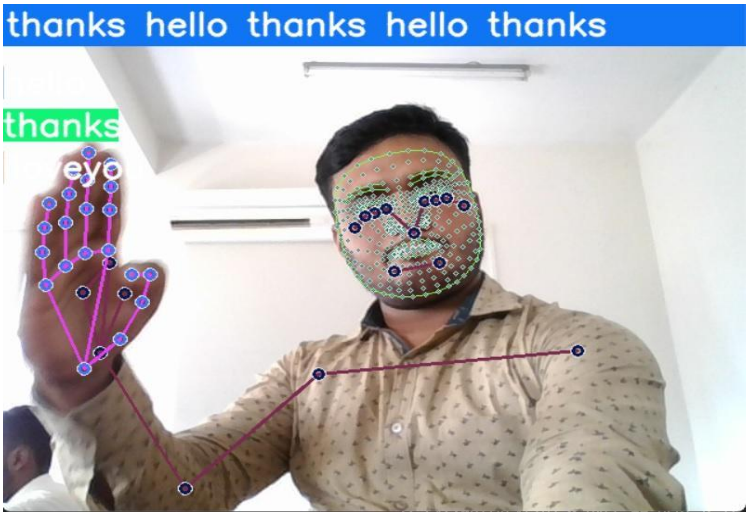


Fig 6. Real time representaion of the model

## VI. COMPARISON WITH SIMILAR TECHNOLOGY AND TERMS

We compare the dataset's efficiency with our method statistically to mpholistic(), mediapipe(), and Faceposters() to identify the quality with other approaches that are closely related to our work. On various datasets, nevertheless, we have used very basic but effective data model to collect the data alongside train the model and test the model on real world.

### *A. Effectiveness quantitative comparison*

As shown in Table 2, our technique greatly outperforms approaches [11, 31] when assessed using the identical train/test data and frame-level metric. The retrieval-based method used by assumed Value. [11] in particular repeats a large number steps which we need to repeat gradually, F1 score is 88.00%, the highest ever reported. All these actions are condensed into tiny segments to prevent tedious segment to action matching. The online lstm 71% F1 score matches these templates with data segments.

TABLE 2. COMPARISON WITH STATE-OF-THE ART APPROACHES ON THE DATASET

| | data [min] | data [min] | time [h] | F. ext. [ms] | Annot. [ms] | Total [ms] | $F_1$ [%](round) |
|---|---|---|---|---|---|---|---|
| Mullar online data | 31 | 40 | 0 | 0.68 | 3.39 | 4 | 71 |
| Mullar ofline data | 25 | 35 | 0 | 0.98 | 0.35 | 2.03 | 89 |
| assumed Value | 21 | 61 | 2 | 9.18 | 0.58 | 7.6 | 88 |
| Online LSTM BF GL | 21 | 61 | 5 | 7.15 | 0.16 | 0.19 | 87 |
| Ofline LSTM BF GL | 21 | 61 | 4.5 | 2.56 | 0.8 | 0.14 | 81 |

### *B. Qualitative comparison*

Very effective techniques are the **LSTM** algorithm[24, 37], which produce improved algorithm results difficult Kinect datasets with low frame rates and large tracking error rates. Both strategies make use of extremely complex, multi-layered architecture. Two modules, one for classification and one for beginning and ending detection, are utilised in [24] and both require multi-stage training. In [37], two attention modules are also utilised. Instead, we suggest a quick and simple method that simultaneously algorithm classification into a single **LSTM** cell. Mainly we offer multilabel output while [22, 38] only allows for the annotation of a single action class at a time.

On the MSRC-12 dataset [14], model-based techniques [6, 41, and 46 provide very competitive results; however, the published F1 scores are computed on operational level.

## VII. CONCLUTION

We have successfully created a system with the help of Long Short-Term Memory which is a part of the advanced Neural Networks. We have observed that the results that is provided by our algorithm is more accurate then RNN, CNN and KNN. Specially the online and offline LSTM a have provided more accuracy to the system and both the algorithms complements each other very efficiently. Our model improves the accuracy of the lengthy data that is created by the online and offline LSTM. The accuracy of detection of moments of our model is near .99 or the detection of the model is 99%.

The data collection rate of our model is near .13 milliseconds and output stream generation of our data is .08 milliseconds which is faster than the models that are present in today’s environments. This algorithm can be implemented with drone camaras to improve accuracy and precession of the camera or focus of the camera alongside it will help us to create the detection systems more accurate.

In future we will combine both the online and offline LSTM so that we can increase the accuracy of the system and more accurate results will be processed. We are very hopeful the targeted system that we have created it will help us to create a virtual environment for the specially abled persons so that they can have a good use of the technology in future.

## VIII. REFERENCES

[1] Fabio Carrara1 · Petr Elias2 · Jan Sedmidubsky, Pavel Zezula2 , LSTM-based real-time action detection and prediction in human motion streams, Received: 4 September 2018 / Revised: 15 May 2019 / Accepted: 23 May 2019 / © Springer Science+Business Media, LLC, part of Springer Nature 2019

[2] Liangliang Cao; Zicheng Liu; Thomas S. Huang, Cross-dataset action detection, Published in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 10.1109/CVPR.2010.5539875, 13-18 June 2010.

[3] Bharat Singh U. of Maryland bharat@cs.umd.edu Tim K. Marks Michael Jones Oncel Tuzel Mitsubish Electric Research Labs (MERL)merl.com Ming Shao Northeastern University mingshao@ccs.neu.edu, A Multi-Stream Bi-Directional Recurrent Neural Network for Fine-Grained Action Detection.

[4] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S. Davis, David J. Crandall, Temporal Recurrent Networks for Online Action Detection, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 5532-5541.

[5] Yanghao Li, Cuiling Lan, Junliang Xing, Wenjun Zeng, Chunfeng Yuan & Jiaying Liu. Online Human Action Detection Using Joint Classification-Regression Recurrent Neural Networks, Conference paper First Online: 16 September 2016, book series (LNIP,volume 9911).

[6] C.S. Pinhanez; A.F. Bobick, Human action detection using PNF propagation of temporal constraints, Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231), 25-25 June 1998.

[7] Xiaojiang Peng & Cordelia Schmid, Multi-region Two-Stream R-CNN for Action Detection, book series (LNIP,volume 9908).

[8] Young Hwi Kim, Seonghyeon Nam, Seon Joo Kim, Temporally smooth online action detection using cycle-consistent future anticipation, Elsevier Volume 116, August 2021, 107954.

[9] Limin Wang, Yu Qiao & Xiaoou Tang, Video Action Detection with Relational Dynamic-Poselets, book series (LNIP,volume 8693).

[10] Zhu L, Shen J, Xie L, Cheng Z (2017) Unsupervised visual hashing with semantic assistant for content-based image retrieval. IEEE Trans Knowl Data Eng 29 (2):472–486.

[11] Zitnick CL, Dollár P (2014) Edge boxes: Locating object proposals from edges. In: Proceedings of the 13th European Conference on Computer Vision, pp 391–405. Springer.

[12] Zhang D, Meng D, Han J (2017) Co-saliency detection via a self-paced multiple-instance learning framework. IEEE Trans Pattern Anal Mach Intell 39 (5):865–878.

[13] Yu G, Yuan J (2015) Fast action proposals for human action detection and search. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 1302–1311. IEEE.

[14] Yan Y, Ricci E, Subramanian R, Liu G, Sebe N (2014) Multitask linear discriminant analysis for view invariant action recognition. IEEE Trans Image Process 23:5599–5611.

[15] Xiang Y, Alahi A, Savarese S (2015) Learning to track: online multi-object tracking by decision making. In: 2015 IEEE international conference on computer vision (ICCV), pp 4705–4713. IEEE.

[16] Wang H, Schmid C (2013) Action recognition with improved trajectories. In: 2013 IEEE international conference on computer vision (ICCV), pp 3551–3558. IEEE.

[17] Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. Int J Comput Vis 104(2)171.

[18] Tian Y, Sukthankar R, Shah M (2013) Spatiotemporal deformable part models for action detection. In: 2013 IEEE conference on computer vision and pattern recognition (CVPR), pp 2642–2649. IEEE.

[19] Rodriguez MD, Ahmed J, Shah M (2008) Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: 2008 IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8. IEEE.

[20] Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems 28 (NIPS 2015), pp 91–99. Curran Associates, Inc.

[21] Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28:976–990.

[22] Aberman K, Wu R, Lischinski D, Chen B, Cohen-Or D (2019) Learning character-agnostic motion for motion retargeting in 2d. ACM Trans Graph 38(4). arXiv:1905.01680

[23] Asadi-Aghbolaghi M, Clape´s A, Bellantonio M, Escalante HJ, Ponce-Lo´pez V, Baro´ X, Guyon I, Kasaei S, Escalera S (2017) A survey on deep learning based approaches for action and gesture recognition in 1The code to reproduce the experiments is publicly available at https://github.com/fabiocarrara/mocap image sequences. In: 2017 12th IEEE international conference on automatic face gesture recognition (FG 2017), pp 476–483

[24] Baltrusˇaitis T, Ahuja C, Morency L (2019) Multimodal machine learning: a survey and taxonomy. IEEE Trans Pattern Anal Mach Intell 41(2):423–443

[25] Barbicˇ J, Safonova A, Pan JY, Faloutsos C, Hodgins JK, Pollard NS (2004) Segmenting motion capture data into distinct behaviors. In: Proceedings of graphics interface 2004. Canadian Human-Computer Communications Society, pp 185–194

[26] Barnachon M, Bouakaz S, Boufama B, Guillou E (2014) Ongoing human action recognition with motion capture. Pattern Recogn 47(1):238–247

[27] Boulahia SY, Anquetil E, Multon F, Kulpa R (2018) Cudi3d: curvilinear displacement based approach for online 3d action detection. In: Computer vision and image understanding

[28] Butepage J, Black MJ, Kragic D, Kjellstrom H (2017) Deep representation learning for human motion prediction and classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6158–6166

[29] Cao Z, Simon T, Wei S, Sheikh Y (2017) Realtime multi-person 2d pose estimation using part affinity fields. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 1302–1310

[30] Chen C, Jafari R, Kehtarnavaz N (2017) A survey of depth and inertial sensor fusion for human action recognition. Multimed Tools Appl 76(3):4405–4425

[31] Du Y, Wang W, Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition. In: 2015 IEEE conference on computer vision and pattern recognition, pp 1110–1118

[32] Elias P, Sedmidubsky J, Zezula P (2017) A real-time annotation of motion data streams. In: 19th International symposium on multimedia. IEEE Computer Society, pp 154–161

[33] Evangelidis G, Singh G, Horaud R (2014) Skeletal quads: human action recognition using joint quadruples. In: 22nd International conference on pattern recognition (ICPR 2014), pp 4513–4518

[34] Field M, Stirling D, Pan Z, Ros M, Naghdy F (2015) Recognizing human motions through mixture modeling of inertial data. Pattern Recognit 48(8):2394–2406

[35] Fothergill S, Mentis H, Kohli P, Nowozin S (2012) Instructing people for training gestural interactive systems. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI '12. ACM, New York, pp 1737–1746

[36] Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

[37] Hussein ME, Torki M, Gowayyed MA, El-Saban M (2013) Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations. In: Joint conference on artificial intelligence (IJCAI 2013), pp 2466–2472

[38] Jain A, Zamir AR, Savarese S, Saxena A (2016) Structural-rnn: deep learning on spatio-temporal graphs. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 5308–5317

[39] Kadu H, Kuo CCJ (2014) Automatic human mocap data classification. IEEE Trans Multimedia 16(8):2191–2202

[40] Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412:6980

[41] Kratz L, Smith M, Lee F (2007) Wiizards: 3d gesture recognition for game play input. In: Proceedings of the 2007 conference on future play. Future play '07, pp 209–212

[42] Kru¨ger B, Vo¨gele A, Willig T, Yao A, Klein R, Weber A (2017) Efficient unsupervised temporal segmentation of motion data. IEEE Trans Multimedia 19(4):797–812

[43] Lakens D (2010) Movement synchrony and perceived entitativity. J Exp Soc Psychol 46(5):701–708

[44] Laraba S, Brahimi M, Tilmanne J, Dutoit T (2017) 3d skeleton-based action recognition by representing motion capture sequences as 2d-rgb images. Comput Anim Virtual Worlds 28(3–4)

[45] Li Y, Lan C, Xing J, Zeng W, Yuan C, Liu J (2016) Online human action detection using joint classification-regression recurrent neural networks. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer vision—ECCV 2016. Springer International Publishing, Cham, pp 203–220

[46] Li K, He FZ, Yu HP, Chen X (2017) A correlative classifiers approach based on particle filter and sample set for tracking occluded target. Appl Math–A Journal of Chinese Universities 32(3):294–312

[47] Li K, He FZ, Yu HP (2018) Robust visual tracking based on convolutional features with illumination and occlusion handing. J Comput Sci Technol 33(1):223–236

[48] Li S, Li K, Fu Y (2018) Early recognition of 3d human actions. ACM Trans Multimedia Comput Commun Appl 14(1s):20:1–20:21