Cognitive Computing and Internet of Things, Vol. 73, 2023, 34–44
https://doi.org/10.54941/ahfe1003972

AI-Powered Real-Time Analysis of Human Activity in Videos via Smartphone

Rico Thomanek[1], Benny Platte[2], Matthias Baumgart[3], Christian Roschke[2], and Marc Ritter[2]

[1]Media Science, Hochschule Mittweida, Germany
[2]Applied Computer and Biological Science, Hochschule Mittweida, Germany
[3]Research Department, Hochschule Mittweida, Germany

ABSTRACT

A major focus in computer vision research is the recognition of human activity based on visual information from audiovisual data using artificial intelligence. In this context, researchers are currently investigating image-based approaches using 3D CNNs, RNNs, or hybrid models with the goal of learning multiple levels of representation and abstraction that enable fully automated feature extraction and activity analysis based on them. Unfortunately, these architectures require powerful hardware to achieve the highest possible real-time processing, which makes them difficult to deploy on smartphones. However, many video captures are increasingly made with smartphones, so immediate classification of the human activities performed and their labeling already during the video capture would be useful for a variety of use cases. However, this requires an efficient system architecture to perform real-time analysis despite limited hardware power. This contribution addresses the approach of skeleton-based activity recognition on smartphones, where the motion vectors of the detected skeleton points are analyzed for their spatial and temporal expression. In this approach, the 3D bone points of a detected person are extracted using an AR framework and their motion data is analyzed in real time using a self-trained RNN. This purely numerical approach enables time-efficient real-time processing and activity classification. This system makes it possible to recognize a person in a live video stream recorded with a smartphone and classify the activity performed. By successfully deploying the system in several field tests, it can be shown that the described approach both works in principle and can be transferred to a resource-constrained mobile environment.

Keywords: Artificial intelligence, Computer vision, RNN, Pose estimation, Human activity analysis

INTRODUCTION

Human activity recognition in videos has been an important topic in computer vision for several decades. The literature reports numerous works on human motion analysis in multimodal data. Human activity detection can be realized with data types in the form of color data (RGB), color and depth data (RGBD), and human body models (Weinland et al. 2011; Ullah 2018). A subdivision of approaches to human activity recognition within this data

is made into traditional manually created features combined with machine learning and holistic deep learning algorithms (Zhang et al. 2019). Although good success has been achieved using traditionally generated features, these hand-generated features require a great deal of human effort and expertise to develop effective feature extraction procedures. Furthermore, recent research studies have shown that traditional methods based on manual features are not suitable for all types of datasets (Al-Faris et al. 2020). Considering this, the deep learning approach, also known as hierarchical learning or deep structured learning, which is based on the concept of Artificial Neural Networks (ANNs), is becoming increasingly popular for human activity recognition. Corresponding models have already been successfully applied in the fields of speech recognition, audio recognition or image processing, where their performance is either superior or comparable to other algorithms. Deep Learning methods allow the automatic processing of raw image and video data for feature extraction, description and classification.

Compared to feature-based approaches, these methods prove to be more powerful and generalizable, but often require data-intensive training to build the models, as they aim to learn multiple representation and abstraction levels that allow a fully automated feature extraction process (Beddiar et al. 2020). With respect to the different source data, Deep Learning methods are divided into (i) image-based and (ii) skeleton-based approaches for activity recognition (Fu et al. 2019). Image-based methods use the original available frames or depth images of the raw video data. Skeleton-based approaches, on the other hand, use the skeletal information used to encode the trajectories of human body joints previously obtained from pose extraction. Commonly used network architectures for activity recognition are Deep Neural Networks (DNN), Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN), and Hybrid Models (Fu et al. 2019; Khan and Ghani 2021; Koohzadi and Charkari 2017).

RELATED WORK

Early methods for activity classification based on DNNs are presented based on previously extracted features from multimodal wearable sensors (Vepakomma et al. 2015). Other researchers (Walse et al. 2016) are similarly using Principal Component Analysis (PCA), a statistical technique that allows the information content of large datasets to be mapped using a smaller set of summary indices, for feature selection from mobile sensor data and a DNN for activity learning.

Krizhevsky et al. trained CNNs for the first time on a sufficiently large image dataset (ImageNet) consisting of over 15 million labeled images (Krizhevsky et al. 2012). The impressive results have initiated a new era for the use of CNNs in activity recognition. Initial research on the use of Convolutional Neural Networks (CNN) for activity recognition based on a single image architecture using a 2D-CNN (2D-ConvNet) was conducted by Karpathy et al. (Karpathy et al. 2014). However, for capturing temporal dynamics within a short period of time, the use of a 3D convolution that considers

multiple consecutive frames is more intuitive for direct hierarchical representation of spatiotemporal data. The 3D convolution is achieved by transferring a kernel to a cube and moving it in three directions over a stack of several consecutive images. The resulting output is a 3-dimensional volume space. An approach to activity recognition using such 3D CNNs was introduced by Ji et al. (Ji et al. 2013). Researchers finally present a network architecture consisting of eight convolutional layers, five pooling layers, and two fully connected layers (Tran et al. 2015). While spatial and temporal observation in 3D CNNs is performed within one network, multiple stream networks take the approach of processing different input streams through separate networks and then merging the individual results (Simonyan and Zisserman 2014). For different input streams, for example, RGB data, stacked optical flow data, extracted trajectories, spectograms of audio signals or depth information are used. For different input streams, for example, RGB data, stacked optical flow data, extracted trajectories, spectograms of audio signals or depth information are used (Kong and Fu 2018; Girdhar et al. 2017). In addition to CNNs, Recurrent Neural Networks (RNNs) represent another network architecture that has been used. RNNs are primarily designed to perform tasks that require a temporal component for sequential information processing, which makes them a good option for activity recognition. However, because RNNs suffer from the vanishing gradient problem, which causes layers to consequently stop learning, they can only be used for short memory tasks (Mueller and Massaron 2020). Long Short-Term Memory (LSTM) networks attempt to counter this problem by introducing a distinction between short- and long-term states in the RNN architecture, allowing for deeper temporal resolution of sequences (Hochreiter and Schmidhuber 1997; Zhang et al. 2017).

Consequently, LSTM networks represent the most popular RNN architecture for capturing long-term temporal dynamics in the context of activity recognition (Zhang et al. 2017). Thus, in activity recognition using recorded sensor data, an LSTM has been used to capture performed motion information and its dependencies (Murad and Pyun 2017; Zebin et al. 2018). LSTMs can thus also be used in the context of skeleton-based activity analysis by considering the spatial and temporal motion information of human bone points as features (Du et al. 2015; Liu et al. 2018; Song et al. 2018). This requires the prior extraction of skeletal data, which is mainly performed using RGBD data. LSTMs are also used in the context of image-based analysis. For example, visual content based on BoW as well as SIFT features has been used to classify activities in soccer sequences (Baccouche et al. 2010). The increasing popularity of neural networks for pose extraction (e.g., (Cao et al. 2019; Wang et al. 2019)) now enables skeleton extraction based on RGB data and thus can be used as a basis for activity analysis using LSTM (Ramirez et al. 2022).

SYSTEM ARCHITECTURE AND WORKFLOW

The skeleton-based activity analysis makes it possible to automatically detect and classify movement patterns of persons. One of the challenges of

skeleton-based activity analysis using smartphones is the extraction of joint points from an image. This requires the use of deep neural networks capable of recognizing human poses and bone points in real time. The current iPhone 14 Pro smartphone enables extraction of human joint points in live video images at up to 60 fps. Joint coordinates can be processed in either 2D or 3D coordinates. While the 2D-based activity analysis provides the coordinates in x- and y-direction, the 3D-based activity analysis additionally provides the depth information. As the bone points are provided as numerical coordinate points, direct and time-efficient processing using LSTM is possible to find cross-correlations of the body keypoints over a variable time frame. Figure 1 shows the workflows for creating the necessary training data set and the classification of activities.

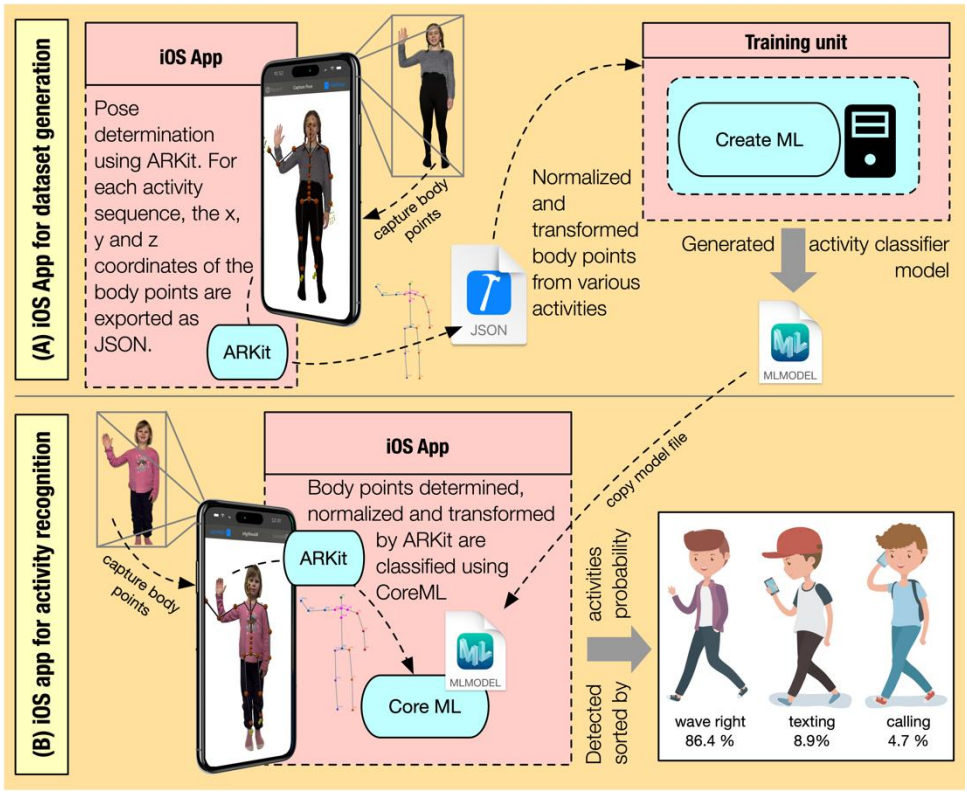

Figure 1: (A) Workflow for creating a training dataset. (B) Workflow for classifying activities in live camera images.

Pose Extraction

Apple's current framework ARKit (ARKit | Apple Developer Documentation n.d.) now enables the extraction of 3D keypoints, which can lead to an increase in detection performance due to the additional third dimension. For further processing, the joint coordinates must be provided relative to a central zero point of the person. This makes the data independent of the person's positioning in the camera image. A total of 18 keypoints are extracted, which

are processed by an LSTM-based activity classifier based on the normalized bone points.

Dataset Generation for Activity Classifier Training

The underlying system architecture is the activity classifier unit presented in the workflow of Thomanek et al. (Thomanek et al. 2020). The activity classifier unit enables the generation of a classification model. The dataset needed for training is generated by ourselves using iPhone 14 Pro. This enables the recording of normalized and transformed 3D body points, which are subsequently used for the training of an LSTM-based activity classifier. The training data is composed of the ten activities shown in Table 1, where multiple activity sequences are recorded for each activity. An activity sequence corresponds to a time-limited record of a motion pattern of a performed activity. An activity sequence of length 3s results in 1620 data points at a recording rate of 30 fps and 18 extracted keypoints. We recorded 50 activity sequences for each activity. Consequently, this results in 81,000 data points to represent one activity. Each data point contains the three coordinates for the x-, y- and z-direction.

Table 1. Everyday activities used for training.

Activity identifier	Description: A Person…
BendOver	… leans forward to pick up something, for example
CallingPhoneWithLeft	… makes a phone call with the left hand
CallingPhoneWithRight	… makes a phone call with the right hand
EatWithRight	… eats or drinks with the right hand
HoldingBox	… carries a larger object with both hands
Nothing	… stands with hands hanging down
SittingDown	… sits down on a chair
TextingPhone	… using a smartphone with both hands
WaveLeft	… waves with the left arm
WaveRight	… waves with the right arm

Training the Activity Classifier

The training data is exported by the app in JSON format shown in Figure 1 (A). For training the activity classifier, we use the CreatML framework provided by Apple (Create ML | Apple Developer Documentation n.d.). This allows the creation of a DeepConvLSTM model (Ordóñez and Roggen 2016) based on recurrent convolutional and LSTM units, allowing the processing as well as fusion of multimodal numerical data points to model temporal dynamics. For training, we used all 18 body points.

Activity Recognition

The real-time activity analysis on the smartphone is analog to the creation of the training data required for the training, shown in Figure 1 (B). The extracted body keypoints are taken from the live image of the integrated

camera and processed directly on the device. This includes the transformation and normalization of the extracted body keypoints. Every Input Vektor of keypoints include previous time-steps as additional part of model input. For example, an input window of size 30 include the current time-step along with 29 previous sets of keypoints.

According to the specifications of the trained DeepConvLSTM model (window size), the motion of each body keypoint is collected in a separate array over the runtime corresponding to the window size. We use the same recording rate of 30 fps to extract the 18 body keypoints and pass them to the classifier. If the number of keypoints defined by the window size has been sampled over the runtime for each of the 18 body keypoints, all data points are passed to the activity classifier for activity prediction. Body keypoints that cannot be extracted by pose detection for example due to occlusions are estimated as far as possible. If no keypoints can be extracted, the input vector for each keypoint is reset to provide correct data collection when keypoints become available again. On successful transfer of the data points to the activity classifier, the activity classifier returns the class label with the highest probability value and a dictionary with several class labels and their probability value.

DISCUSSION

The activity data recorded for training were at a length of 3s per activity sequence and a frame rate of 30 fps. We selected the following 10 daily activities shown in Table 1 and used them for training.

For training the activity classifier, we experimented with different iterations and window sizes. Iterations were performed at values of 10, 50, 70, 140, 300, and 600. For the window sizes, we used the values 15, 30, 45, and 60, resulting in 24 trained models with different precision and recall values for the mentioned activities.

The results of the experiments are shown in Figure 2 and Figure 3. The results show that the classification works very well for some activities depending on the combination of iteration and window size (e.g. “BendOver”, “SittingDown” with many Precision and Recall values of 1), while it is less successful for other activities (e.g. “CallingPhoneWithLeft” with Precision and Recall of 0).

It indicates that using more iterations and a larger window size can improve total classification performance. For example, using 50 iterations and a window size of 15 results in lower performance compared to using 70 iterations with a window size of 45. However, there are also exceptions to this rule, such as the classification of “WaveLeft”, which often deteriorates with more iterations. In general, it can be stated that a too high iteration value of 600 again leads to a worse total performance due to overfitting on training data.

As can be seen in Figure 3, we were able to achieve best results at an iteration of 300 and a window size of 60, which means that at 30 fps two seconds of an activity sequence are used for analysis.

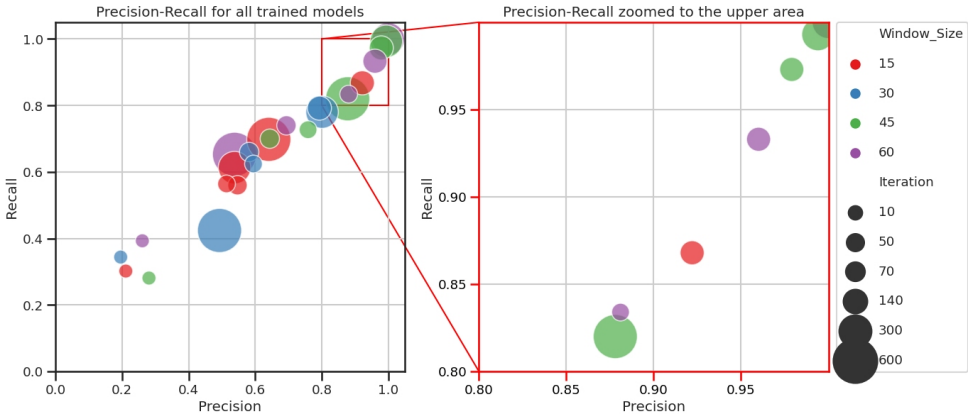

Figure 2: Validation results of all trained models. depicted are the macro average precision vs. macro average recall.

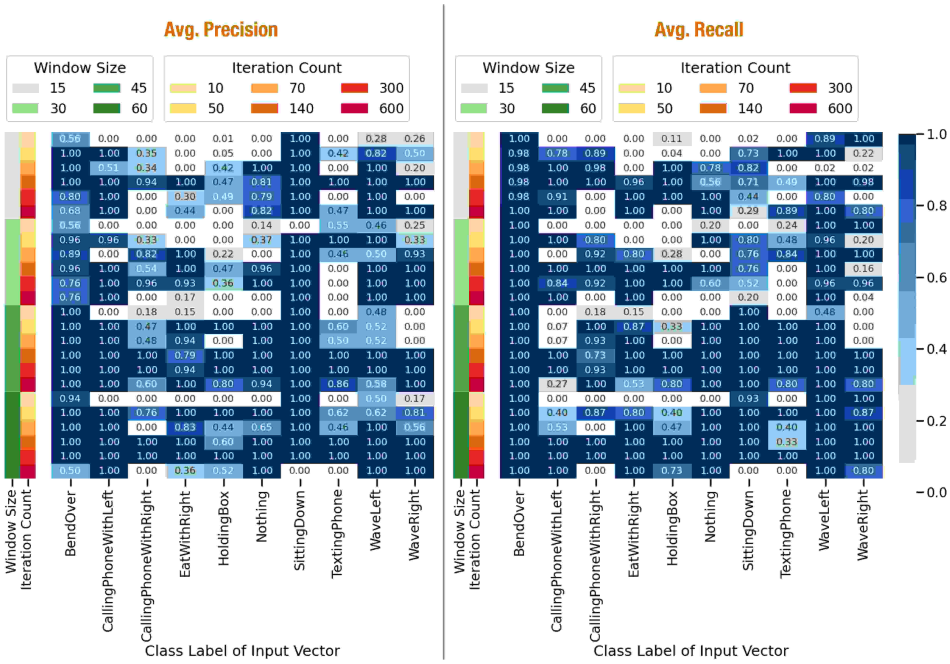

Figure 3: Average of precision and recall divided by class vs. iteration and window size.

The generated model has a size of 1.8 MB and can be used directly in the source code of the iOS app. The individual body points are passed to the classification model. The size of the array corresponds to the Window Size. To establish independence, the body proportions are normalized.

From the age of six, body proportions differ only slightly from each other (Schünke et al. 2018). The skeleton-based activity analysis can thus be used for children and adults similarly, without the need to use special anthropometric training data to create the models. Since the data is normalized relative to body proportions, the app allows classification independent of body size.

Figure 4 shows the confusion matrix for the model trained with 300 iterations and a window size of 60. Here it becomes obvious that especially the activities CallingPhoneWithRight, HoldingBox, Nothing and TextingPhone show deficits in the classification. This could be due to the similarity of the skeletal movements (e.g. TextingPhone and Holdingbox).

true label \ predicted label	Bend Over	Calling Phone With Left	Calling Phone With Right	Eat With Right	Holding Box	Nothing	Sitting Down	Texting Phone	Wave Left	Wave Right
BendOver	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
CallingPhoneWithLeft	0.000	0.892	0.000	0.000	0.002	0.000	0.000	0.016	0.090	0.000
CallingPhoneWithRight	0.000	0.000	0.767	0.219	0.000	0.000	0.000	0.000	0.000	0.014
EatWithRight	0.000	0.000	0.012	0.910	0.078	0.000	0.000	0.000	0.000	0.000
HoldingBox	0.000	0.000	0.000	0.247	0.729	0.014	0.000	0.010	0.000	0.000
Nothing	0.000	0.000	0.000	0.148	0.139	0.713	0.000	0.000	0.000	0.000
SittingDown	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
TextingPhone	0.000	0.019	0.000	0.062	0.243	0.014	0.000	0.662	0.000	0.000
WaveLeft	0.000	0.027	0.000	0.000	0.000	0.000	0.000	0.000	0.973	0.000
WaveRight	0.000	0.000	0.076	0.077	0.000	0.000	0.000	0.000	0.000	0.847

Figure 4: Confusion matrix for the detection of the ten activities listed in Table 1 using our own dataset.

CONCLUSION

The presented method enables real-time based activity detection in live video images using a smartphone at a frame rate of 30 fps. Due to the resulting quotient of window size and frame rate (window_size/fps) of, for example, 60 and 30 fps, there is a time offset of 2s to ensure the required data buffering before forwarding to the activity classifier. The results obtained in the experiments could be confirmed in a field test with test persons in the age range of 13 - 45.

Overall, it can be said that the classification performance is good for most activities, but there is still room for improvement, especially for the activities with low Precision and Recall values. One possible method to improve the classification performance here could be to limit the number of body points used to the most necessary ones. Currently, we use all 18 body points for classification. Considering upper body and lower body activities separately could potentially increase classification performance.

Many activities are similar in terms of their skeletal movements. Eventually involved objects can provide additional contextual information for the performed activity here (e.g. TextingPhone: person holding smartphone vs. ReadingBookperson reading a book). The additional use of an object classifier and the associated inclusion of the involved objects as features of the activity classifier could potentially increase recognition performance.

REFERENCES

Al-Faris, M., Chiverton, J., Ndzi, D. and Ahmed, A. I. (2020). A Review on Computer Vision-Based Methods for Human Action Recognition. *Journal of Imaging* [online], 6(6), p. 46.

ARKit | Apple Developer Documentation. Available from: https://developer.apple.com/documentation/arkit/ [accessed 27 January2023].

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C. and Baskurt, A. (2010). Action Classification in Soccer Videos with Long Short-Term Memory Recurrent Neural Networks BT - Artificial Neural Networks – ICANN 2010. In: Diamantaras, K., Duch, W., and Iliadis, L. S., eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 154–159.

Beddiar, D. R., Nini, B., Sabokrou, M. and Hadid, A. (2020). Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, 79(41), pp. 30509–30555. Available from: https://doi.org/10.1007/s11042-020-09004-3.

Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S.-E. and Sheikh, Y. A. (2019). Open Pose: Real time Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, pp. 1–1.

Create ML | Apple Developer Documentation. Available from: https://developer.apple.com/documentation/createml/ [accessed 30January 2023].

Du, Y., Wang, W. and Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1110–1118.

Fu, M., Chen, N., Huang, Z., Ni, K., Liu, Y., Sun, S. and Ma, X. (2019). Human action recognition: A survey. In: *Lecture Notes in Electrical Engineering*. Springer Verlag, pp. 69–77. Available from: https://link.springer.com/chapter/10.1007/978-981-13-7123-3_9 [accessed 25 June 2020].

Girdhar, R., Ramanan, D., Gupta, A., Sivic, J. and Russell, B. (2017). Action VLAD: Learning Spatio-Temporal Aggregation for Action Classification. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 3165–3174.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, 9(8), pp. 1735–1780. Available from: https://doi.org/10.1162/neco.1997.9.8.1735.

Ji, S., Xu, W., Yang, M. and Yu, K. (2013). 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), pp. 221–231.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. (2014). Large-Scale Video Classification with Convolutional Neural Networks. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1725–1732.

Khan, N. S. and Ghani, M. S. (2021). A Survey of Deep Learning Based Models for Human Activity Recognition. *Wireless Personal Communications*, 120(2), pp. 1593–1635. Available from: https://doi.org/10.1007/s11277-021-08525-w.

Kong, Y. and Fu, Y. (2018). Human Action Recognition and Prediction: A Survey., 28 June 2018. Available from: https://arxiv.org/abs/1806.11230 [accessed 5 June 2020].

Koohzadi, M. and Charkari, N. M. (2017). Survey on deep learning methods in human action recognition. *IET Computer Vision*, 11(8), pp. 623–632. Available from: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cvi.2016.0355.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Image Net Classification with Deep Convolutional Neural Networks. In: Pereira, F. et al., eds. *Advances in Neural Information Processing Systems*. Curran Associates, Inc. Available from: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Liu, J., Shahroudy, A., Xu, D., Kot, A. C. and Wang, G. (2018). Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Mueller, J. P. and Massaron, L. (2020). *Deep Learning kompakt fürdummies*. 1. Auflage. Weinheim: Wiley.

Murad, A. and Pyun, J.-Y. (2017). Deep Recurrent Neural Networks for Human Activity Recognition. *Sensors*, 17(11). Available from: https://www.mdpi.com/1424-8220/17/11/2556.

Ordóñez, F. and Roggen, D. (2016). Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* [online], 16(1), p. 115. Available from: https://www.mdpi.com/1424-8220/16/1/115 [accessed 4 February 2020].

Ramirez, H., Velastin, S. A., Aguayo, P., Fabregas, E. and Farias, G. (2022). Human Activity Recognition by Sequences of Skeleton Features. *Sensors*, 22(11), p. 3991. Available from: https://app.dimensions.ai/details/publication/pub.1148169383.

Schünke, M., Schulte, E., Schumacher, U., Voll, M. and Wesker, K. H. (2018). *PROMETHEUS Allgemeine Anatomie und Bewegungs system*.

Simonyan, K. and Zisserman, A. (2014). Two-Stream Convolutional Networks for Action Recognition in Videos. *Advances in Neural Information Processing Systems* [online], 1(January), pp. 568–576. Available from: https://arxiv.org/abs/1406.2199 [accessed 13 February 2020].

Song, S., Lan, C., Xing, J., Zeng, W. and Liu, J. (2018). Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection. *IEEE Transactions on Image Processing*, 27(7), pp. 3459–3471.

Thomanek, R., Roschke, C., Zimmer, F., Rolletschke, T., Manthey, R., Vodel, M., Platte, B., Heinzig, M., Eibl, M., Hosel, C., Vogel, R. and Ritter, M. (2020). Real-Time Activity Detection of Human Movement in Videos via Smartphone Based on Synthetic Training Data. In: *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision Workshops, WACVW 2020*.

Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M. (2015). Learning Spatio temporal Features with 3D Convolutional Networks. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. pp. 4489–4497.

Ullah, J. (2018). *Human Action Recognition and Localization in Videos* [unpublished]. National University of Computer & Emerging Sciences.

Vepakomma, P., De, D., Das, S. K. and Bhansali, S. (2015). A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities. In: *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. pp. 1–6.

Walse, K. H., Dharaskar, R. v and Thakare, V. M. (2016). *PCA Based Optimal ANN Classifiers for Human Activity Recognition Using Mobile Sensors Data BT*. Satapathy, S. C. and Das, S., eds. Cham: Springer International Publishing.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W. and Xiao, B. (2019). Deep High-Resolution Representation Learning for Visual Recognition. *CoRR*, abs/1908.0. Available from: https://arxiv.org/abs/1908.07919.

Weinland, D., Ronfard, R. and Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 2011.

Zebin, T., Sperrin, M., Peek, N. and Casson, A. J. (2018). Human activity recognition from inertial sensor time-series using batch normalized deep LSTM

recurrent networks. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. pp. 1–4.

Zhang, H. B., Zhang, Y. X., Zhong, B., Lei, Q., Yang, L., Du, J. X. and Chen, D. S. (2019). A comprehensive survey of vision-based human action recognition methods. *Sensors (Switzerland)*, 2019.

Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S. and Li, Z. (2017). A Review on Human Activity Recognition Using Vision-Based Method. Park, D. S., ed. *Journal of Healthcare Engineering*, 2017, p. 3090343. Available from: https://doi.org/10.1155/2017/3090343.