Real-Time Activity Detection of Human Movement in Videos via Smartphone Based on Synthetic Training Data

Rico Thomanek[1], Tony Rolletschke[1], Benny Platte[1], Claudia Hösel[1], Christian Roschke[1], Robert Manthey[1], Manuel Heinzig[1], Richard Vogel[1], Frank Zimmer[1], Matthias Vodel[1], Maximilian Eibl[2], Marc Ritter[1]

[1] University of Applied Sciences Mittweida, D-09648 Mittweida, Germany
{firstname.lastname}@hs-mittweida.de
[2] Chemnitz University of Technology, D-09107 Chemnitz, Germany
{firstname.lastname}@informatik.tu-chemnitz.de

Abstract

Current research in the domain of video activity detection focuses on real-time activity detection. This includes multiple approaches in the mobile environment, such as the detection of correct motion sequences in the sports and health area or in safety-relevant environments. Current trends focus on the use of 3D CNNs. This work describes a approach to combine the results of a human skeleton point detector with an LSTM on mobile devices. Frameworks for pose detection are combined with LSTMs for activity detection with sensor data, optimized for the mobile area. The resulting system allows the direct detection of a person pose and the classification of activities in a video recorded with a smartphone. The successful application of the system in several field tests shows that the described approach works in principle and can be transferred to a resource-limited mobile environment by optimization.

1. Introduction

The detection of human activity in videos is becoming increasingly important. One focus is the detection of activity in real-time. Especially in the mobile environment there are numerous fields of application. For example, smartphones could be used in the sports and health sectors to detect correct movements. But they could also be used for safety-relevant movements. For example, body cams could start recording a video only when critical activities are performed. While current approaches use tubes and streams in the form of 3D CNNs [4, 16, 8], we focus on [17, 10, 9, 5], which detects activities based on the motion constellation of skeletal points or sensor values using an LSTM architecture. In our approach, however, we use the 2D results of a human keypoint detector [2] and process them using a unidirectional LSTM [12]. As ground truth we use synthetically generated animations from which the positions of the bone points are extracted. Networks for pose recognition can already reach 15 FPS on mobile devices, and LSTM networks have been used for many years for activity recognition using sensor data. For activity detection the detected body points of the pose recognition are transformed and normalized as 2 dimensional point values and then interpreted by the LSTM. For pose recognition a pre-trained net was used. The LSTM itself was developed using the framework Turicreate [7]. For each activity the training material consists of 50 synthetically generated videos of 10 seconds each.

2. System Architecture and Workflow

The developed system enables the generation of an activity classifier and its usage for the asymmetric analysis of human activities in videos. The basis for the training and the activity recognition are pose information of human motion sequences, which are provided in the form of body keypoints. For activity analysis we use Long Short-Term Memory (LSTM) networks to find cross correlations of body keypoints over a variable time frame. The trained model can then be used on both computing nodes and mobile devices. The underlying system architecture we use is described by Thomanek et. al. [14, 13]. This architecture offers, due to the available standardized interfaces to the interprocess communication, the possibility to exchange distributed process instances or add further process instances without having to adapt the entire system architecture. In this context, the process chain shown in Figure 1 is realized. As can be seen, the system structure enables the administration and usage of training, validation and test data, which are referenced in a common database and stored in a central data storage. For easy usability, the *Video Segmentation Unit* offers the possibility to split the video material into their frames, store them centrally via Common Internet File System (CIFS)/Server Message Block (SMB) and reference them in the database. The frames referenced in the database are then loaded by the *Pose detection and transformation unit* using CIFS/SMB and passed to *OpenPose*. Based on this, the determined body key points are personalized by normalization and transformation. If several persons are involved, an additional person tracking is executed. The resulting dimensionless body points are stored in the database with the original *OpenPose* data. The *Activity classifier unit* enables the creation of an activity classifier and the activity classification for selected test data based on the imported data.

2.1. Synthetic activity animations as basis for pose extraction

The implementation of the 3D environment as well as all necessary 3D objects and animations takes place directly in the game engine Unity®, because the required functionalities are already integrated. Unity® is a runtime and development environment for the development of interactive applications by Unity Technologies, headquartered in San Francisco. [15]. With Unity, the 3D scene shown in Figure 2 is used, which contains a generalized human character created by Paul Chen [3]. The scene is deliberately designed to be simple and the character is placed at the coordinate origin. The character consists of a visible mesh and an invisible skeleton to transfer activity animations to the character. The animations are from Adobe's platform Mixamo [1] and include *Talking_phone*, *Texting_phone*, *Loading*, *Closing*, *Opening*, *Open_Trunk*, *Closing_Trunk*, *Carrying*, *Transport_HeavyCarry*, *Unloading*, *Waving* and *Nothing*. To create the ground truth, for each activity 5 different animations are placed on the character one by one and 10 different videos from multiple positions are recorded. The cameras used are positioned at an angle of 45 degrees above the character and focus on the zero point of the scene. Each recording has a duration of 10 seconds. For each activity, 50 videos are generated with a sampling rate of 30 FPS and a resolution of 1280x720.

2.2. Generation of the ground truth based on the data of the pose extraction

For the detection of person keypoints we use OpenPose in the *Pose detection and transformation unit* with the output parameter *write_json*. This gives us an associated JSON file for each frame, which contains a people array of all detected people. For the activity determination we use the 2D keypoints, which contain for each detected body point's the x- and y-coordinate as well as its confidence value. We use the 18 body points of the COCO Keypoint Dataset. To import the determined keypoints, we use a JSON parser implemented with Python, which not only imports the data into the database, but also performs a plausibility check, trans-

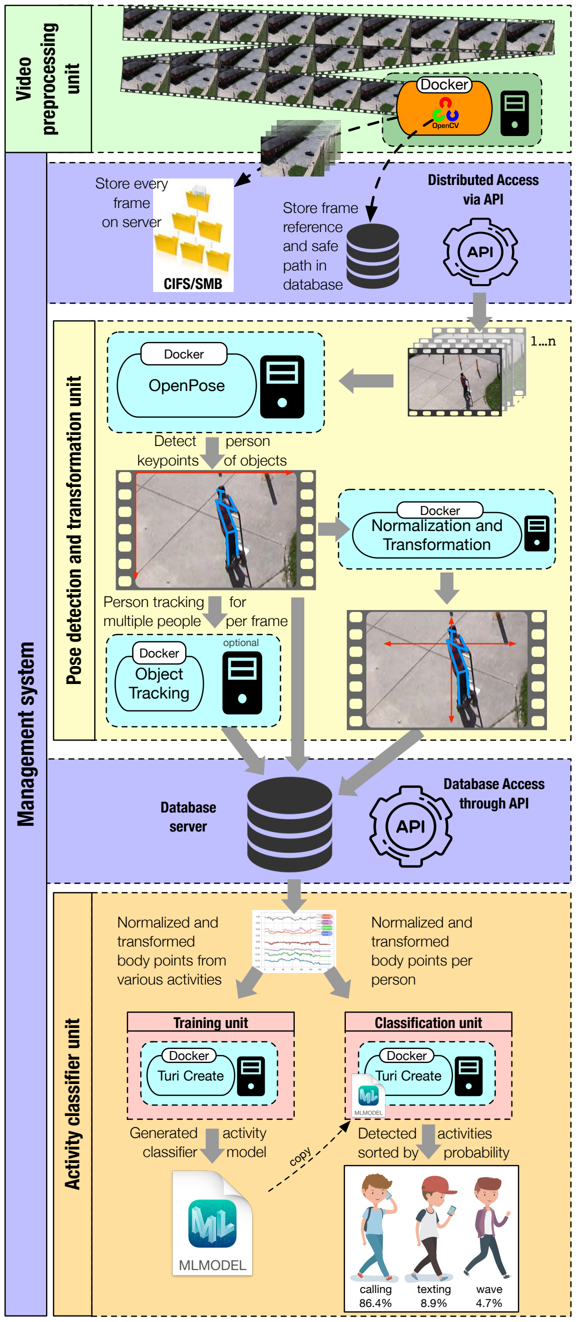
Video preprocessing unit
Docker
OpenCV
Management system
Store every frame on server
CIFS/SMB
Store frame reference and safe path in database
Distributed Access via API
API
Pose detection and transformation unit
1...n
Docker
OpenPose
Detect person keypoints of objects
Docker
Normalization and Transformation
Person tracking for multiple people per frame
optional
Docker
Object Tracking
Database server
Database Access through API
API
Activity classifier unit
Normalized and transformed body points from various activities
Normalized and transformed body points per person
Training unit
Docker
Turi Create
Classification unit
Docker
Turi Create
MLMODEL
Generated activity classifier model
copy
Detected activities sorted by probability
MLMODEL
calling 86.4%
texting 8.9%
wave 4.7%

Figure 1. Workflow of the proposed system.

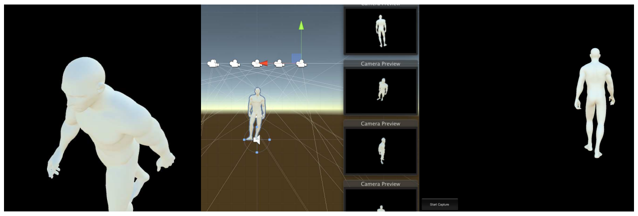
Figure 2. Exemplary representation of the development tool for the creation of ground truth using synthetic data.

formation and normalization. The supplied coordinates of the body points refer to the coordinate origin of the frame image, which is located in the upper left corner of the image. In order to recognize a person-centralized movement, all keypoints must therefore be transformed to a new coordinate origin. For this we have chosen the neck point of the detected person. Furthermore, the x- and y-values detected by OpenPose depend on the resolution of the image. In order to ensure activity detection with other video resolutions, all coordinates must be standardized. We have defined the distance between neck and hip as one normalization range and scaled all other body points to the associated value range. Figure 3 illustrates this fact.

2.3. Data storage and management

The original OpenPose data and the values standardized and transformed by us are then stored in the database in separate database tables. The plausibility check guarantees that only the OpenPose values are taken into the database where at least arms, neck, shoulders and hips have been recognized. In the case that the imported OpenPose data is test data, a tracking algorithm must then be used to track people. For this purpose, we have used the tracking algorithm developed by Platte et. al. [11]. The implemented database schema is shown in Figure 4. In the database table *resultsActivityClassifier* the object-related detected activities are stored under specification of the video ID, frame ID and object ID.

2.4. Creation of the LSTM-based activity classifier

The activity classifier is created based on the normalized and transformed body points created using *Turi Create*. The underlying model of the activity classifier in Turi Create is able to recognize temporal patterns in the moving body points. It is based on convolutional layers to extract temporal characteristics from a single prediction window, e.g. an upward curved motion of the arm and elbows may be a strong indicator of the activity *Talking_phone*. It also uses recurring layers to extract temporal features over time. For the creation of the activity classifier we have implemented our own Python class. Each normalized and transformed body point is regarded by us as a two-dimensional sensor. Since each keypoint is described by an x- and y-coordinate, we get 36 individual sensor values. The function for creating the activity classifier of *Turi Create* expects the data values assigned to an activity to be sorted chronologically in ascending order. *Turi Create* allows the import of the Ground Truth from a database using SQL query. Using this SQL query, the body points to be used for training can be easily defined and determined in chronological order. The basic architecture is shown in figure 5.

Furthermore, the definition of the size of the prediction window and the number of iterations to be performed during the training process must be transferred. The prediction window defines the time intervals between the predictions and the number of body point samples to consider for the prediction. For the *Prediction_Window* we have chosen the value 15. At a frame rate of 30 FPS, this results in a prediction frequency of 2Hz.

2.5. Implementation of a mobile application for real-time activity detection

The actvity classification unit of our system enables real-time activity detection with smartphones. For this purpose we use the current ARKit framework [6] from Apple for pose extraction. ARKit enables a real-time capable pose extraction on mobile devices. We use a window size of 15 images to estimate the activity. The extracted body points are then passed to a self-implemented class and transformed and normalized according to the principle of 2.2. The transformed and normalized 2D coordinates are then passed as an array to the generated model whereupon an activity classification is performed. The iOS app and the corresponding model have been published at https://github.com/hsmwfgie/ios_realtime-activity-classificator.

3. Results

For the creation of the activity classifier, synthetically generated activity animations were created. Since each animation represents only one person, the body points extracted by *OpenPose* do not need to be subjected to additional person tracking. The extracted Normalized and Transformed body points are then used for training. The resulting activity classifier can be used within the *Classification unit*, but also in mobile devices for activity determination. The use of the *Classification unit* causes the storage of the detected activities in the database. These can be evaluated directly or merged with other methods for activity analysis. Due to the proportionality of human bodies and the dimensionless activity classifier, activities of persons of different body sizes can be detected equally.

The practical proof of real-time activity determination using smartphones was provided in the form of a mobile application for iOS devices on *GitHub*. The published model contains the activities *Talking_phone*, *Waving* and *Nothing*.

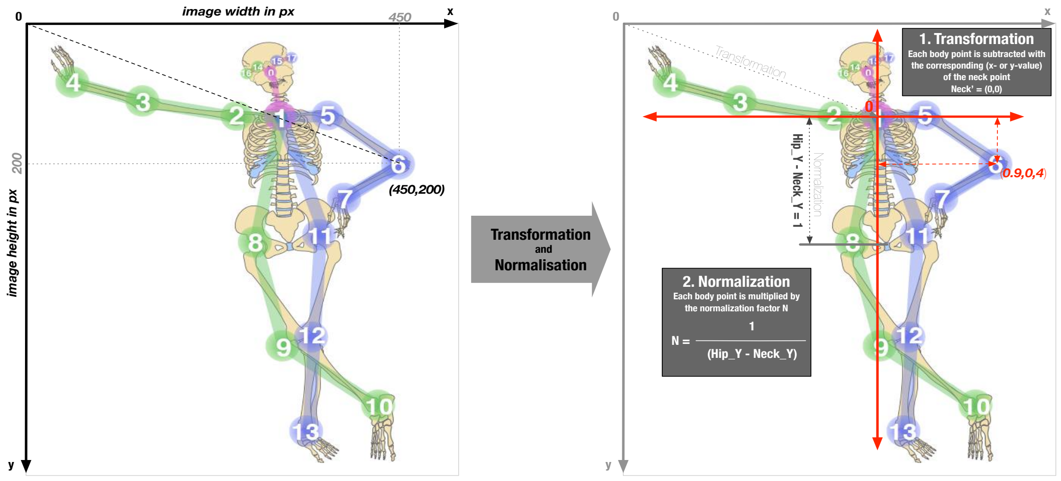

Figure 3. Normalization and transformation of the OpenPose results into an object-related coordinate system.

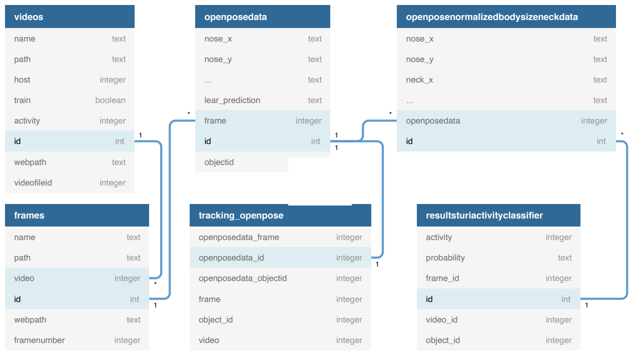

Figure 4. Database schema for referencing videos, frames and classifying activities.

4. Conclusion and Future Work

The presented system environment enables the creation and use of an activity classifier based on pose information. The processing rate of 150 FPS enables real-time classification of a single person on mobile devices. Currently, only asynchronous activity classification is possible using the system environment described. In future work, the real-time classification for several persons will be implemented via the developed infrastructure. To determine the ground truth, synthetic activity animations are currently generated using OpenPose and subsequent normalization and transformation. In future work, however, these data will be read directly during synthesis in the game engine Unity® and stored in the database.

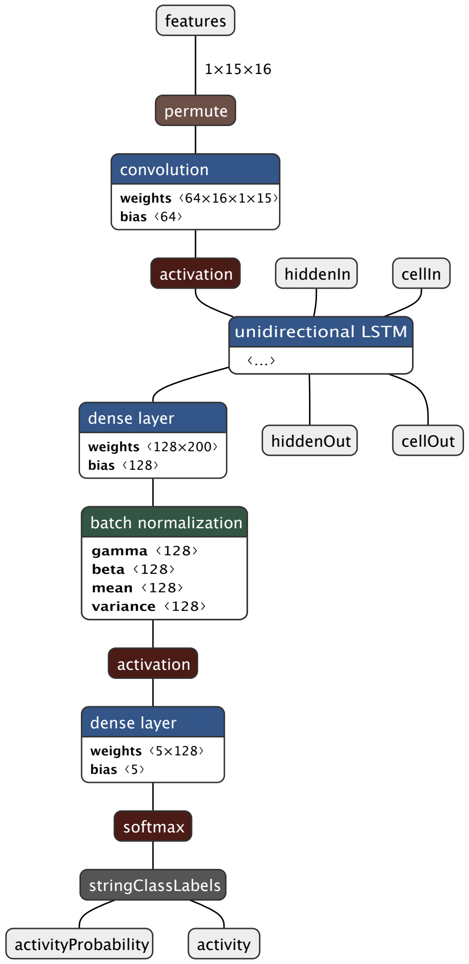

Figure 5. Architecture of the LSTM based activity classifier.

References

[1] Adobe. Mixamo, 2019.

[2] Z. Cao, G. Hidalgo Martinez, T. Simon, S.-E. Wei, and Y. A. Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[3] P. Chen. Männliche Base Mesh 3D-Modell, 2015.

[4] G. Gkioxari and J. Malik. Finding action tubes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 759–768, 2015.

[5] Y. Guan and T. Plötz. Ensembles of Deep LSTM Learners for Activity Recognition using Wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2017.

[6] A. Inc. ARKit — Apple Developer Documentation.

[7] A. Inc. turicreate: Turi Create simplifies the development of custom machine learning models, 2017.

[8] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Action Tubelet Detector for Spatio-Temporal Action Localization. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 4415–4423, 2017.

[9] C. Li, P. Wang, S. Wang, Y. Hou, and W. Li. Skeleton-based action recognition using LSTM and CNN. In *2017 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2017*, 2017.

[10] J. Liu, G. Wang, L. Y. Duan, K. Abdiyeva, and A. C. Kot. Skeleton-Based Human Action Recognition with Global Context-Aware Attention LSTM Networks. *IEEE Transactions on Image Processing*, 2018.

[11] B. Platte, R. Thomanek, C. Roschke, and M. Ritter. Person Tracking and Statistical Representation of Person Movements in Surveillance Areas. *INTERNATIONAL JOURNAL OF DESIGN, ANALYSIS AND TOOLS FOR INTEGRATED CIRCUITS AND SYSTEMS, VOL. 7, NO. 1, OCTOBER 2018*, pages 20–25, 2018.

[12] A. Sherstinsky. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. aug 2018.

[13] R. Thomanek, B. Platte, C. Roschke, R. Manthey, T. Rolletschke, C. Hösel, M. Ritter, and F. Zimmer. Use of Multiple Distributed Process Instances for Activity Analysis in Videos. pages 320–327. 2019.

[14] R. Thomanek, C. Roschke, B. Platte, R. Manthey, T. Rolletschke, M. Heinzig, M. Vodel, F. Zimmer, and M. Eibl. A scalable system architecture for activity detection with simple heuristics. In *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision Workshops, WACVW 2019*, pages 27–34, 2019.

[15] J. Thorn, Alan; Doran, John P.; Zucconi, Alan; Palacios. *Complete Unity 2018 Game Development - Explore techniques to build 2D/3D applications using real-world examples*. Packt Publishing Ltd, Birmingham, 2019.

[16] J. Zhao and C. G. M. Snoek. Dance with Flow: Two-in-One Stream Action Detection. *arXiv.org*, cs.CV, 2019.

[17] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-Occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016.