Automated Annotation of Sensor data for Activity Recognition using Deep Learning

Maik Benndorf, Frederic Ringsleben, Thomas Haenselmann and Bharat Yadav[1]

Abstract: Within this work-in-progress, we aim to automate the annotation of Sensor data for generating training data for Activity Recognition (AR) of multiple persons. Usually, the activities are executed and recorded from test persons under the supervision of an instructor, which may influence in many cases the natural behaviour of the test persons and the authenticity of the data. In this work, we suggest how this influence can be reduced and how the Sensor data can be annotated automatically by using video capturing, openpose[Ca17] for extracting human key points and a neuronal network to classify the activities. By automatically annotating the selected activities we show the feasibility of our approach.

Keywords: Activity Recognition; Automated Annotation; Sensor data; Openpose; Deep Learning

1 Introduction

Generating training data for activity recognition is a time consuming process. In many cases, activities are executed and recorded under the instructions of a supervisor. The work of the supervisor starts with the definition of the activities. Afterwards, he has to instruct the test person and supervise the whole process of recording. During this process, he has to create a protocol and after the recording, he has to annotate the activities usually manually. Therefore, the effort involved, increases with the number of test persons.

A side effect of this approach is, that the test persons do not behave naturally during the recording since they were guided all the time. One solution to this problem is to provide only general conditions to these test persons and let them act nearly independently. However, this may lead to a huge quantity of sensed data that has to be annotated manually.

With this work in progress, we aim to address an approach for automated annotation of sensor data of multiple persons, where the effort in annotating and the influence to the natural behaviour of the test persons should be minimized. Our approach is a conceptual process chain. Therefore, we use the approach where only basic conditions were provided to the test persons. For example, the instructor could tell the test persons: Please walk around, go upstairs and wait there a while. Thereby, the test persons are filmed by a camera. This

[1] University Of Applied Sciences Mittweida, Faculty of Applied Computer Sciences & Biosciences, Mittweida 09648, Germany {benndorf, ringsleb, haenselm, byadaf}@hs-mittweida.de

doi:10.18420/in2017_220

video serves as a protocol of what has been done at which time. On the other hand, we utilize this video to extract the activities of the test persons. We show the feasibility of our approach by one test person and three selected activities (standing, walking and walking upstairs) recorded in three different perspectives (front, side and back). Therefore, this work is organized as follows. The section 2 provides an overview of what has been done in this area. Section methodology (3) will describe our approach in more detail. Section 4 summarizes this work and finally provides an outlook on further work and future research.

2 Related Work

With this approach we left the area of signal processing and touched the area of multimedia annotation. In this case, we have to distinct between the annotation of images (human pose recognition) and the annotation of image sequences or videos (human activity recognition). A lot of work has been done in human pose recognition, shown in numerous publications like [An14; Ca17; Ra06; ROR08]. Therefore, basically the same strategies were used. The first part is to extract body from the image. Afterwards, the patterns thus obtained can be compared with known patterns in order to derive the human pose. In order to extract body parts from images there are different approaches. A relatively old one is based on limbs detection by using the Curvature Scale Space (CSS) [MM86]. In [RS06], they trained deformable models and use a conditional random field to determine the position of the body parts. One recent way in order to extract the positions of key-points is the Convolutional Pose Machines (CPM)[We16]. A Convolutional Neural Network (CNN) is trained to key-points like ankle, knee and hip. After the body parts have been extracted, they can be compared with known human poses. Therefore, multiple Datasets like [An14; KH09; Sz16] can be used.

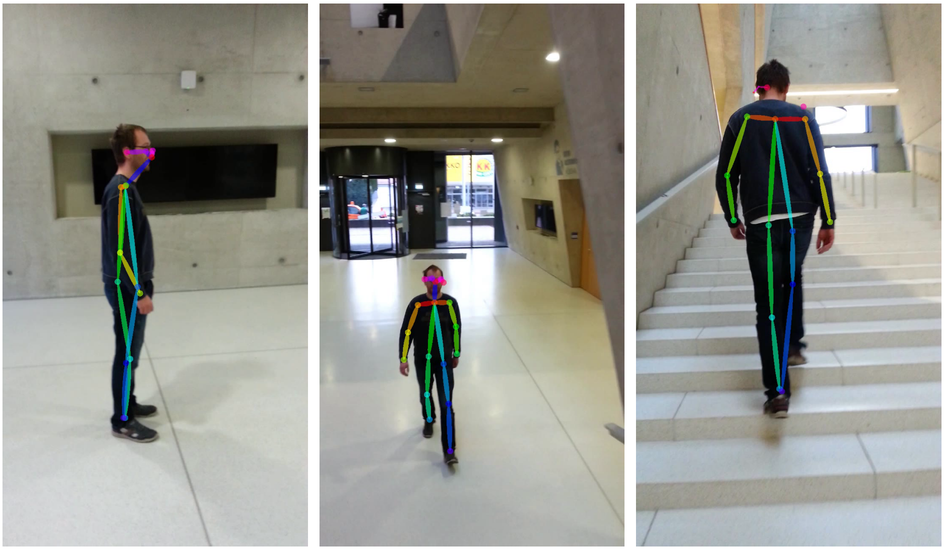
This approach is also often used in images sequences or videos. So for example Openpose[Ca17] is a library for real-time multi-person key-point detection and multi-threading written in C ++ using OpenCV and Caffe[Ji14]. This is a further development of authors of CPM. Nevertheless, this library can only be used in order to extract the key-points of an image sequence or video. In order to get the activities from the poses, the poses are linked to each other. Some approaches [BN12; KHC10; KPH05] are using hidden Markov models for this. In [ROR08] they describe the steps to annotated the human movement in a image sequence using a human walk example. Therefore, they used a model of movement. In [PCZ15] the authors shows a way to improve the estimation by using the optical flow between neighbouring frames. In [Fa03] used the CSS to extract the pose. Afterwards, a matrix is created, whereby the transitions of the individual poses are charged with costs. Thus, the cost of the transition between two poses at which the person is initially lying and afterwards is walking are very high. So this problem is turned into an optimization task in which the path with the minimum costs through this matrix is searched. The authors solve this task by means of dynamic programming. The most of these approaches belong to the posed-base methods. In contrast to this are the dense trajectories in [Wa13]. This approach is mainly based on the optical flow and uses for the detection of the human movement. In [CS14; Ke13; Ra16] the extraction of collective activities of multiple persons like “people playing basketball“ is done. Nevertheless, in our opinion, there is no approach that can extract the independent activities of multiple persons.

3 Methodology

Within this work in progress, we would like to show the feasibility of our approach. Therefore, our activities are executed by one test person. In future works, we look forward to expand this to multiple persons. During the recordings, our test person has performed various activities, which were taken from different perspectives (side, front and back). For this work, we decided using three of these activities (standing, walking and walking upstairs). So we could produce more than 60 minutes of video material.

3.1 Extraction of the Key-Points

The first step in our methodology is to extract all the key-points from the video. Therefore, we decided for the Openpose-library (described in section 2).



(a) Standing (side)

(b) Walking (front)

(c) Walking Upstairs (back)

Fig. 1: Annotated Images exported from the resulting Video of Openpose (the perspective is given in brackets)

This library currently supports 18 key-points, for example: neck, hip and ankles. The output of this library is on one hand a video, where the key-points are marked and the associated key-points (e.g. knee and hip) are connected by coloured lines. In Figure 1 example, the output for our test person performing the three activities from different perspectives can be seen. On the other hand, a JSON-file can be exported for each frame within the video. For each recognized person, this file contains all of the 18 key-points in the form of X-Coordinate and Y-Coordinate and the recognition-confidence for the key-point (can be seen in Listing 1).

```
{
"version":0.1,
"people":[{
"body_parts":[
123.45, 678.90 , 0.1234,...
]
}]
}
```

List. 1: Openpose: example JSON-file for one frame

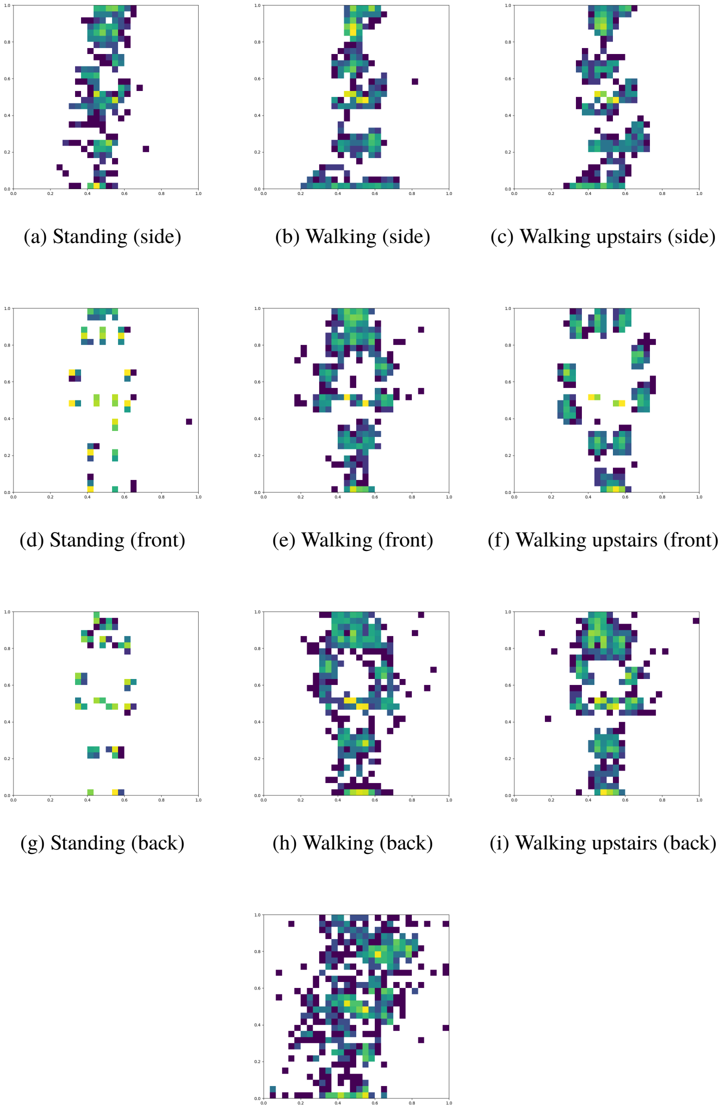
From all those files we created a matrix where all the information for every single frame is stored. In order to make them comparable and size variant, the next step is to normalize all the key-points.

3.2 Normalization of key-points

Since the hip is near to the center of the human body, we decided for taking the hip as a central point for the normalization. The center point X is calculated by by determining the center between the right and the left hip. Subsequently, X is subtracted from all other key points. X is now located at the coordinates $0; 0$. Now we have to determine the absolute maximum value and divide each key-point by two times of the maximum value. Thus, the coordinates of all the key-points are in the range of -0.5 and 0.5. Finally, we have to add 0.5 to bring X to the center of the image. Thus, the hip is now located at the coordinates $0.5, 0.5$ and all other key points are in the range of 0 and 1 in both X and Y direction.

3.3 Build classification images from key points

For the purpose of classification, we need to store the movement patterns of the activities in images. Therefore, we decided to use two dimensional histogram (heatmaps) and a sliding window approach with a window length of 3 seconds. Hence, we generated up to $1,500$ heatmaps for the activities standing, walking, walking upstairs. All other activities are defined as noise. Figure 2 shows examples of the histograms for all the selected activities for the three perspectives and one example image for noise. Finally, our dataset is built up from 1, 500 heatmaps that can be used to classify the activities.



(a) Standing (side) (b) Walking (side) (c) Walking upstairs (side)

(d) Standing (front) (e) Walking (front) (f) Walking upstairs (front)

(g) Standing (back) (h) Walking (back) (i) Walking upstairs (back)

(j) Example Noise

Fig. 2: Heatmaps for the Activities

3.4 Classification

For the purpose of classification, we decided to use a CNN based Lasagne Library built up from the tutorial of [Ka]. We also have used the configuration of this tutorial: The Layers were organized as follows: INPUT, CONV, POOL, CONV, POOL, CONV, POOL, DENSE, DENSE, DENSE. Since the Convolutional Layers expects a 4-dimensional Input Layer the input shape looks like this:

```
INPUT: (None, 3, 64, 64)
```

Whereby, "None" is a placeholder for the number of images passed through the net, the parameter 3 corresponds to the number of dimensions (RGB) and the last two parameters indicate the size of the input image. Since the heatmaps are larger than these $64x64$ pixels, the heatmaps have to be scaled. We used a pools size of 2. The filter size is defined as $3x3$ without padding and the number of filters increases with every following convolution (64, 128, 256).

Hence, we trained our CNN with $1,500$ heatmaps in 100 epochs and split the whole dataset into 70% for training 15% for evaluation and 15% for the purpose of testing.

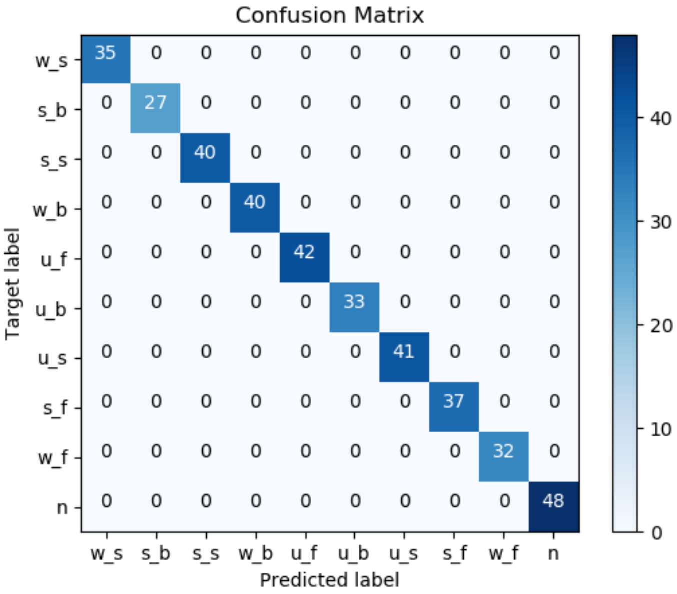


Fig. 3: Confusionmatrix

As it can be seen in Figure 3 we reached a total accuracy of 100 percent, supported by the confusion matrix. This is, of course, an extraordinary result that also raises the question of whether the relatively complex methodology is necessary at all. We believe that this result is due to the fact that we were focused on only one test person and finally only three activities. In addition, we think that our methodology becomes increasingly necessary for our further work (more activities, mor persons). Nevertheless, it can be used as a proof of concept for our approach. Finally, we have to create an annotation file. Therefore, we write the outcome of each image into a textfile which is indexed by the starting frame number. This number can be used in order to map the recognized activity to the timing vector of the sensor data of the smart phone.

4 Conclusion and Outlook

In this work, we could show the feasibility of an automatic annotation of sensor data by one test person, one perspective of video filming and three selected activities. This can be taken as a proof of concept for our future works. The next steps are: we have to consider more activities, more perspectives in video filming and multiple persons in order to create a real automatic annotation tool by multiple persons for sensor data. Finally, we would like to compare the sensor data recorded by this approach with sensor data recorded by the traditionally approach in order to make the general influence on the natural behavior of the test persons visible.

Acknowledgement

This work is based on the ideas of the group pose analysis at the Chemnitz University of Technology. A special thanks to Danny Kowerko and Stefan Kahl for thought-provoking impulses and fruitful discussions.

References

[An14] Andriluka, M.; Pishchulin, L.; Gehler, P.; Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Pp. 3686–3693, 2014.

[BN12] Banerjee, P.; Nevatia, R.: Pose based activity recognition using Multiple Kernel learning. In: Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, pp. 445–448, 2012.

[Ca17] Cao, Z.; Simon, T.; Wei, S.-E.; Sheikh, Y.: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In: CVPR. 2017.

[CS14] Choi, W.; Savarese, S.: Understanding collective activities of people from videos. IEEE transactions on pattern analysis and machine intelligence 36/6, pp. 1242–1257, 2014, URL: http://ieeexplore.ieee.org/abstract/document/6654151/, visited on: 06/30/2017.

[Fa03] Farin, D.; Haenselmann, T.; Kopf, S.; Kühne, G.; Effelsberg, W.: Segmentation and classification of moving video objects. Handbook of Video Databases: Design and Applications 8/, pp. 561–591, 2003.

[Ji14] Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093/, 2014.

[Ka] Kahl, S.: Tutorial: Image Classification with Lasagne – Part 1 – Stefan Kahl Blog, URL: http://medien.informatik.tu-chemnitz.de/skahl/2016/10/20/image-classification-with-lasagne-part-1/, visited on: 06/11/2017.

[Ke13] Ke, S.-R.; Thuc, H. L. U.; Lee, Y.-J.; Hwang, J.-N.; Yoo, J.-H.; Choi, K.-H.: A review on video-based human activity recognition. Computers 2/2, pp. 88–131, 2013, URL: http://www.mdpi.com/2073-431X/2/2/88/htm, visited on: 06/30/2017.

[KH09] Krizhevsky, A.; Hinton, G.: Learning multiple layers of features from tiny images./, 2009.

[KHC10] Kim, E.; Helal, S.; Cook, D.: Human activity recognition and pattern discovery. IEEE Pervasive Computing 9/1, 2010.

[KPH05] Kellokumpu, V.; Pietikäinen, M.; Heikkilä, J.: Human activity recognition using sequences of postures. In: MVA. Pp. 570–573, 2005.

[MM86] Mokhtarian, F.; Mackworth, A.: Scale-based description and recognition of planar curves and two-dimensional shapes. IEEE transactions on pattern analysis and machine intelligence/1, pp. 34–43, 1986.

[PCZ15] Pfister, T.; Charles, J.; Zisserman, A.: Flowing convnets for human pose estimation in videos. In: Proceedings of the IEEE International Conference on Computer Vision. Pp. 1913–1921, 2015.

[Ra06] Ramanan, D.: Learning to parse images of articulated bodies. In: NIPS. Vol. 1. 6, p. 7, 2006.

[Ra16] Ramanathan, V.; Huang, J.; Abu-El-Haija, S.; Gorban, A.; Murphy, K.; Fei-Fei, L.: Detecting events and key actors in multi-person videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Pp. 3043–3053, 2016, URL: http://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Ramanathan_Detecting_Events_and_CVPR_2016_paper.html, visited on: 06/30/2017.

[ROR08] Rogez, G.; Orrite-Uruñuela, C.; del Rincón, J. M.: A spatio-temporal 2D-models framework for human pose recovery in monocular sequences. Pattern Recognition 41/9, pp. 2926–2944, 2008.

[RS06] Ramanan, D.; Sminchisescu, C.: Training deformable models for localization. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Vol. 1, IEEE, pp. 206–213, 2006.

[Sz16] Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261/, 2016.

[Wa13] Wang, H.; Kläser, A.; Schmid, C.; Liu, C.-L.: Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision 103/1, pp. 60–79, 2013.

[We16] Wei, S.-E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y.: Convolutional pose machines. In: CVPR. 2016.