Multi-camera skeleton-based activity recognition

Assisted Living and Health Monitoring

Silvia Vecchietti and Ilario Zamberlan

July 23, 2020

UNIVERSITY OF TRENTO

Contents

1 Introduction

Assisted living and health monitoring are technological services finalized to improve the life quality of the people who need a support in daily life, for example old people, people with disabilities or person in rehab after a surgery. These kind of systems are important and they are going to become increasingly demanded in particular because of the life expectancy is gradually enlarging: it is indispensable to extend the life period in which old people are autonomous. Another important function of these technologies, as said before, is the rehabilitation. For example with these kind of system, for a just discharged hospital patient it is possible to resume motory activity while staying at home.

In these kind of applications information and communication technologies are used, in particular computer vision and artificial intelligence, in order to monitor the daily-life activities and make them safer.

The goal of our project is to realize a computer vision system which, using two or more cameras, is able to define the activities of a person in his home using the 3D person's skeleton. We selected four main activities: sit, stand, walk and eat or drink. Our targets are old people or people in their rehab, so these are the most relevant activities for our purposes. The idea is to obtain, as a output of the system, a list of activities and the respective duration that can be used by the doctor to analyze the patient's situation, without the latter have to leave his home.

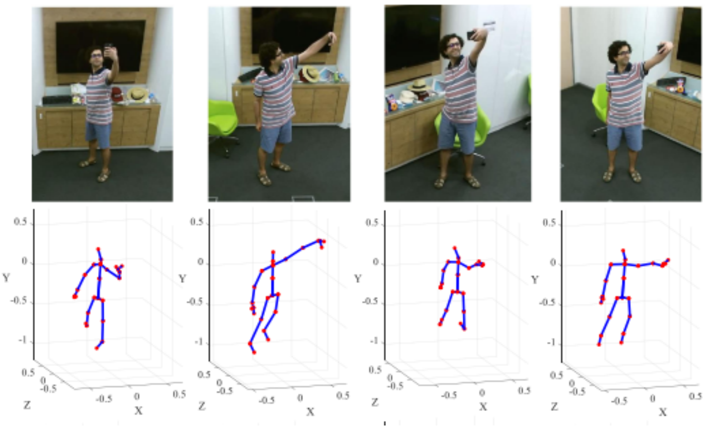
Regarding the realization, we used as a starting point the project [1] in which was implemented a neural network to take over the following actions: stand, walk, run, jump, sit, squat, kick, punch and wave. These starting software didn't give very good results so we tried to improve it by moving from a 2D to a 3D skeleton model and also adding some feature like the angle between joints and body velocity. We also adapt the actions at our application idea. Four actions are considered in this work: walk, stand, have meal and sit.

2 3D model and dataset

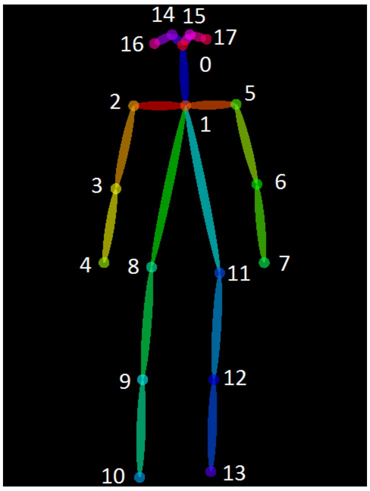
In this section is explained how the algorithm works and it is highlighted the salient points.

2.1 3D model and feature

The main idea of this project is to switch from a 2D to a 3D model in order to increase the performance of the action recognition. This change allows the system to have more precised data and to be not effected to the body rotation.
Using a 2D skeleton, the same action can be seen changing due to different viewpoints as can seen in the figure[2]:



The depicted action is "doing a selfie" and it is the same in all the four photos. If we observe the skeletons below the four matching skeletons seems really different one from the other because of the different rotation. This is a hard problem for action recognition.
Using a 3D model this problem is easily resolved.
Openpose's skeleton format has been used, which has 18 joints and for each joint three coordinates are associated: x, y and z:



To train the neural network we extract the following feature:

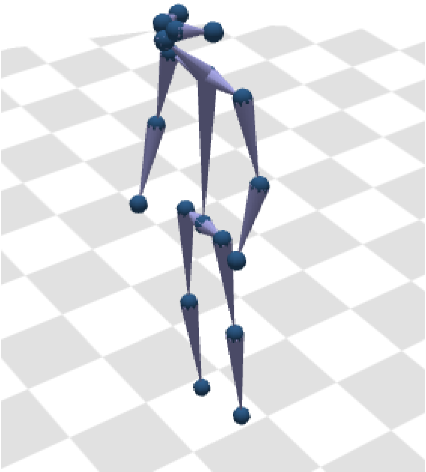
- Joint position
- Joint velocity
- Angle between joints

2.2 Dataset, training and testing

The following dataset has been used for training:

- CMU Panoptic Dataset [3]
- CAD-120 [4]
- Two videos [5] [6]

In the CMU panoptic Dataset the output skeleton is different respect to OpenPose:



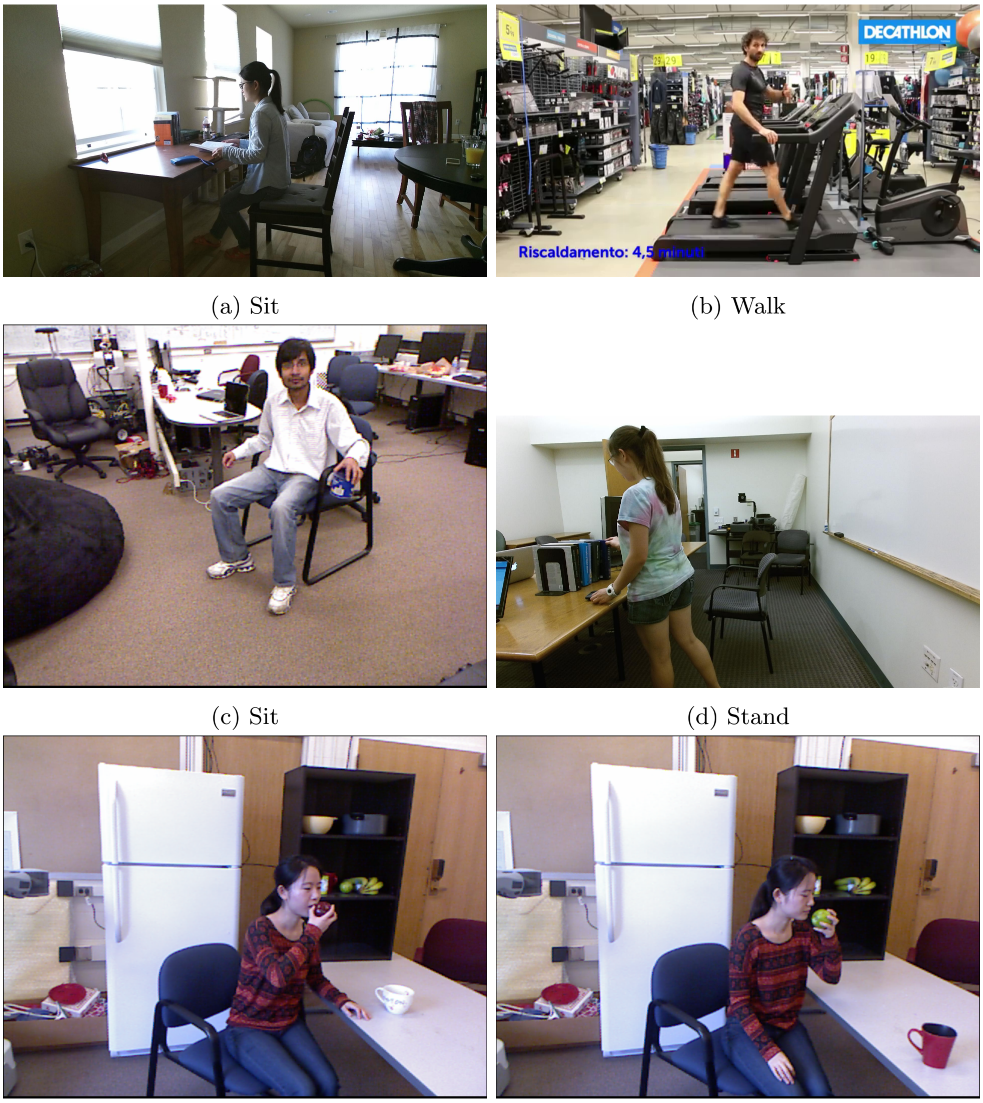
In the other Datasets were present just a 2D picture with a depth map of them. Meaningful action has been selected and converted in 3D using ”Lifting from the deep” [7]. Lifting from the deep generate another skeleton’s format.
Starting from this skeletons it is possible to obtain the OpenPose format applying a simple matching of different indexes.

Index	OpenPose	Panoptic	Lifting from the deep
0	Nose	Neck	Medium(Lhip, Rhip)
1	Neck	Nose	Lhip
2	Rsho	Medium(Lhip, Rhip)	Lkne
3	Relb	Lsho	Lank
4	Rwri	Lelb	Rhip
5	Lsho	Lwri	Rkne
6	Lelb	Lhip	Rank
7	Lwri	Lkne	Chest
8	Rhip	Lank	Neck
9	Rkne	Rsho	Nose
10	Rank	Relb	Head
11	Lhip	Rwri	Lsho
12	Lkne	Rhip	Lelb
13	Lank	Rkne	Lwri
14	Leye	Rank	Rsho
15	Reye	Reye	Relb
16	Lear	Leye	Rwri
17	Rear	Rear	
18	-	Lear	

The main goal was set as an improvement of the original work[1]. For this reason the main structure has been maintained. In particular, the neural network implemented is a multi-layer Perceptron with 3 hidden layers, each of 50 size.
In training phase the code use about seventy percent of the composed dataset for training and the other thirty for test.
As evaluation,a video has been made in which all the four actions(walk, stand, have meal and sit) are performed. Due to a lack of a stereo camera, from the 2D video, ”lifting from the deep” has been used to extract 3D skeletons.

2.2.1 Dataset example

Due to the mixed composition, the obtained dataset is made with picture taken in different scenario. Some samples can be found below



(a) Sit

(b) Walk

(c) Sit

(d) Stand

(e) Meal

(f) Meal

Figure 1: Samples taken from the different dataset mentioned before

3 Results

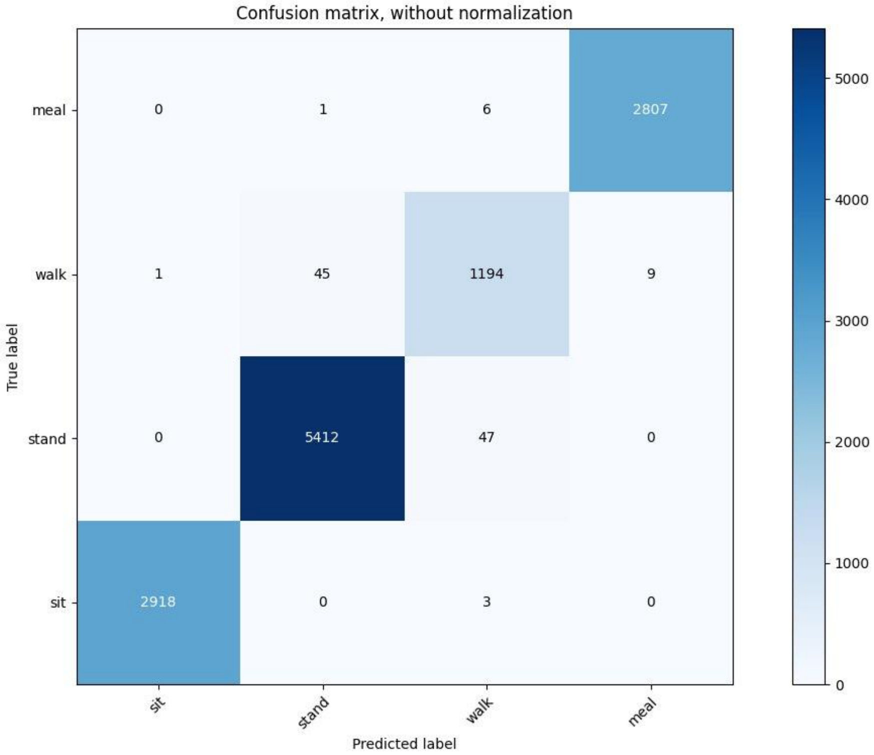
3.1 Training

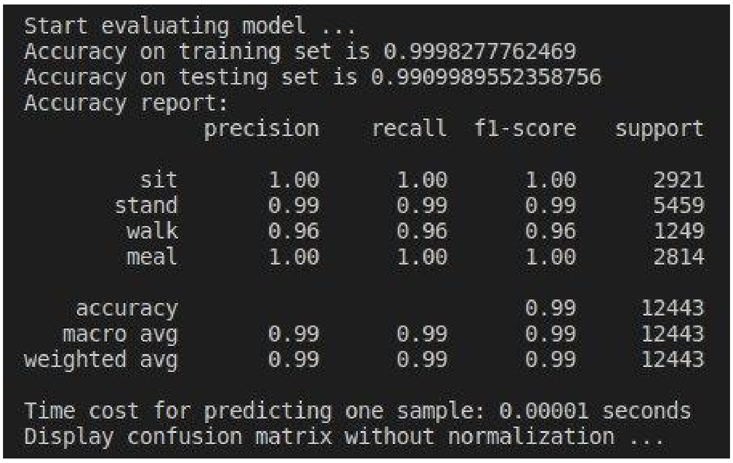
Firstly, the net was trained using a labeled data, in which the following actions were present: walk, stand, have meal and sit.
Number of sample per action:

Action	Number of sample
sit	31265
stand	42633
walk	5937
meal	9587

Table 1: Number of each samples present in the composed dataset

The below images show the confusion matrix and the accuracy report obtained from the training of the multilayer perceptron.





```
Start evaluating model ...
Accuracy on training set is 0.9998277762469
Accuracy on testing set is 0.9909989552358756
Accuracy report:
               precision    recall  f1-score   support

         sit       1.00      1.00      1.00      2921
       stand       0.99      0.99      0.99      5459
        walk       0.96      0.96      0.96      1249
        meal       1.00      1.00      1.00      2814

    accuracy                           0.99     12443
   macro avg       0.99      0.99      0.99     12443
weighted avg       0.99      0.99      0.99     12443

Time cost for predicting one sample: 0.00001 seconds
Display confusion matrix without normalization ...
```

3.1.1 Evaluation

In order to have a better output it has been stabilized with a sliding average. It has been supposed that the action can not change more the once within three seconds.
This is in order to remove impossible sequence like:
walk, walk, walk, walk, walk, walk, walk, walk, walk, sit, sit, walk, walk, walk, walk, walk, walk, walk, walk, walk. This sequence is not possible in a 30 fps video.
The sequence taken as an example would become all flatted as *walk*:

Action	Actually present	Prediction w/o average correction	Prediction with average correction
sit	419	0	0
stand	298	16	0
walk	1076	472	450
meal	869	2138	2182

Table 2: Number of frame with respective action classified in the video evaluation

The percentages of predicted actions are:

Type	Standard output	Flattened output
Right	43.65 %	46.43 %
Miss-classified	56.34 %	53.56 %

Table 3: percentages of predicted actions

It has been noticed that in most of cases the actions sit were miss-classified as *meal*.
So a further trial has been made by replacing the action meal with the action sit, and we performed again the training phase.

3.2 Training without *'meal'*

Number of sample per action:

Action	Number of sample
sit	40852
stand	42633
walk	5937

Table 4: Number of each samples present in the dataset

As in the previous case, the confusion matrix and the accuracy report is shown.

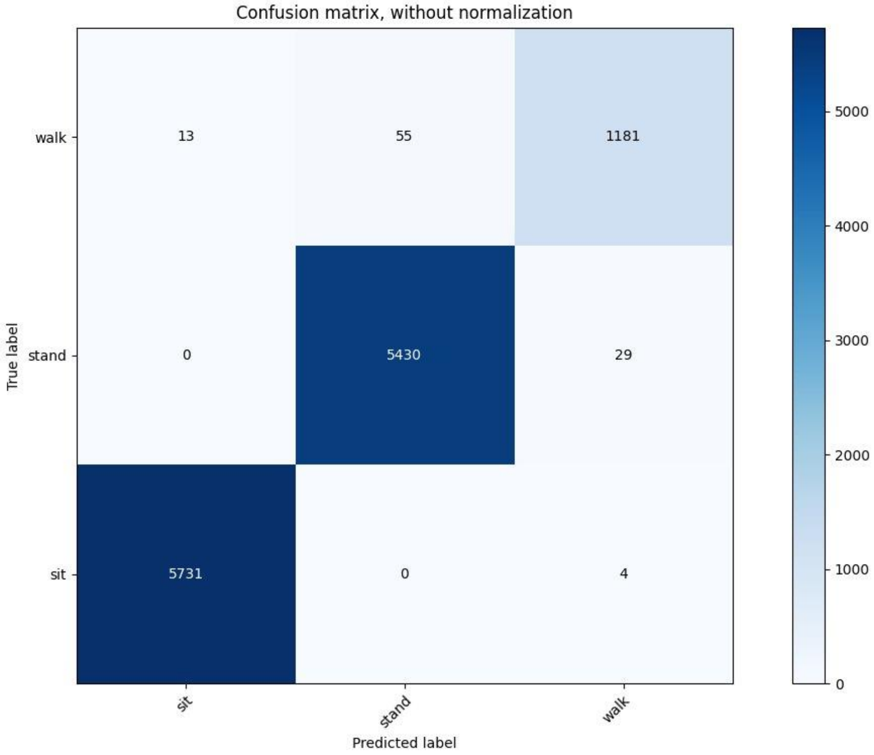
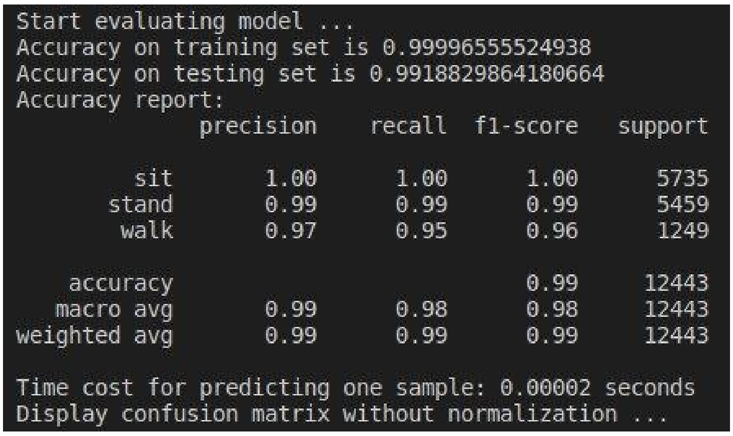


Figure 2: Confusion matrix (without *meal*)


```
Start evaluating model ...
Accuracy on training set is 0.99996555524938
Accuracy on testing set is 0.9918829864180664
Accuracy report:
              precision    recall  f1-score   support

         sit       1.00      1.00      1.00      5735
       stand       0.99      0.99      0.99      5459
        walk       0.97      0.95      0.96      1249

    accuracy                           0.99     12443
   macro avg       0.99      0.98      0.98     12443
weighted avg       0.99      0.99      0.99     12443

Time cost for predicting one sample: 0.00002 seconds
Display confusion matrix without normalization ...
```

Figure 3: Accuracy(without *meal*)

3.3 Evaluation without *'meal'*

To compare this step with the previous phase (the one with *'meal'*) the same flattening post-process has been applied. In the table below it's written the number of actions recognized in the video test.

Action	Actually present	Prediction w/o average correction	Prediction with average correction
sit	1288	2146	2092
stand	298	7	0
walk	1076	474	540

Table 5: Number of frame with respective action classified in the video evaluation

The percentages of predicted actions are:

Type	Standard output	Flattened output
Right	64.16 %	64.17 %
Miss-classified	35.84 %	35.83 %

Table 6: percentages of predicted actions - stabilized

As can be seen from the results performance does not improve removing "meal" action.
The code generates an output video in which it is displayed the predicted label at each frame.

4 Conclusions

The system used do not give excellent results.
The main problem is given by the dataset used. This dataset in fact is not consistent. Except for Panoptic which have 3D skeleton's information for the remaining part of the dataset we started from a 2D image and we obtained a 3D skeleton: this means that the data are not always precised. This let the net to not properly understand the real body shape in the 3D world.
We also have found difficulties to obtain the data we needed. In our dataset the action ”walk” is less present with the respect to the other actions. This also might be a problem.
Furthermore, extracting body joints for *house* activity may land to some difficulties due to the fact there are a lot of occlusion. For example most of the cases a person is sitting, a major part of the body is occluded by the table or desk in front of him. This made more complex building a proper dataset.
Finally we don't take advantage of the fact that inputs frames are correlated over time. We have adapted the original Multilayer Perceptron used in [1] but it is preferable to consider using used a RNN (for example LSTM) which could increase consecutive frame correlation capabilities.

References

[1] felixchenfy, “Realtime-action-recognition.” https://github.com/felixchenfy/Realtime-Action-Recognition/.

[2] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, “View adaptive neural networks for high performance skeleton-based human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[3] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. S. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, “Panoptic studio: A massively multiview system for social interaction capture,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[4] H. S. Koppula and A. Saxena, “Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation,” *Robotics: Science and Systems (RSS)*, 2013.

[5] C. Minusio, “Camminata in salita sul tapis roulant.” `https://www.youtube.com/watch?v=Jn52_g7CKx0`, April 2019.

[6] B. e in forma in 30 minuti, “Allenamento 30 minuti tapis roulant.” `https://www.youtube.com/watch?v=rXjx5Jplfz8`, Dicember 2019.

[7] DenisTome, “Lifting-from-the-deep-release.” https://github.com/DenisTome/Lifting-from-the-Deep-release.