A CNN Based Approach to Detecting Walk-in-place for Virtual Reality

Aniruddha Prithul
University of Nevada Reno
aprithul@nevada.unr.edu

Abstract

Research in action recognition using deep neural networks has seen great progress in recent times. Deep learning techniques are often more robust and accurate than hand crafted techniques. However, its use in the field of virtual reality interaction has been limited. In this paper we analyze the viability of using a deep learning based action recognition technique to detect walk-in-place action for virtual reality locomotion. Our findings suggest that walking-in-place action can be detected with a high level of accuracy and thus is a promising avenue for farther research.

1. Introduction

Action recognition is the process of inferring what action is taking place in a sequence of time series data i.e. a video sequence. This sub-domain of machine learning has seen great improvements in recent years, thanks to the advancements made in deep learning techniques. The usage of action recognition is vast. It can be used in the medical field, for example in fall detection. Or it may be used for surveillance purposes. Another important use of action recognition is in the field of Human Computer Interaction (HCI). In HCI, we are concerned with designing the most effective and intuitive ways of communicating between humans and computers. One such intuitive technique is the use of gestures for interaction. This relies on the successful recognition of the actions performed by the users.

Virtual reality (VR) is a sub-domain of HCI that has seen a resurgence in research in recent times. While a lot of groundbreaking works have been done on many aspects of VR, an open question that still remains is the question of locomotion. Many of the VR experiences attempt to map the limited available physical space to a much larger virtual space. This means that a one to one mapping of physical and virtual body translation is impossible. To solve this issue many artificial locomotion techniques have been proposed i.e. teleportation, head tilt, walking in place etc. Of these, walking in place is deemed to induce a higher level of presence [2], which is highly desirable in VR. Existing approaches of detecting walking in place rely on threshold based approaches where the acceleration changes of a wearable accelerometer [17] is observed to determine whether the user is performing the action. Some approaches rely on observing the vertical displacement of the the user's head [9] Kinect based approaches have also been explored [21] where the joint rotations are observed. While these approach works, it has the potential of generating false positives. For example if the user physically walks to a different point in the tracking space, the generated acceleration may be wrongly interpreted as walking in place by the system. Manually handling all cases might be error prone.

Instead, if an deep learning based action recognition approach is taken where we observer the full body movement of the user, we may be able to generate user inputs in a more reliable and robust manner. This paper is inspired by such possibility and tries to present an alternative approach to implementing walking-in-place.

2. Related Work

Activity recognition is a fast evolving field in the domain of computer vision. Early research involved working with color images for activity recognition. These relied on template matching and state space based approaches[1]. Nowadays it's more common to have approaches utilizing RGB-D images. The depth channel of an RGB-D image can provide important information about the geometry and position of body parts. The availability of cheaper depth enabled cameras have helped promote their usage. Li et al.[10] adopted a bag of points approach to model the human postures. An action graph was then constructed to represent the motion dynamics which used a BLMD decoding scheme for activity recognition. HON4D by Oreifej and Liu [12] extracted 4D normal data from depth images. Their descriptor was used with a SVM to classify actions.

After the success of AlexNet [7], it's become commonplace to extract features using deep learning methods rather than hand crafting them. Rahmani et al.[13] adopted such a strategy where they trained a CNN with synthetic pose data. The dataset was created by applying motion capture data to 3D human models. After training, the classifier was able

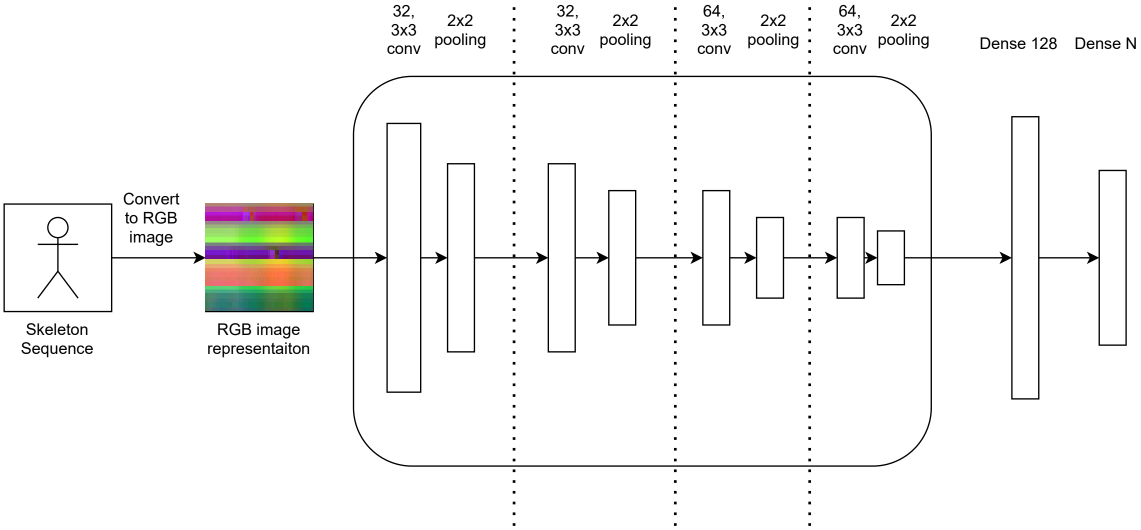

Figure 1. Network overview

to recognize actions from real depth images with considerably better performance than the state-of-the-art. Wang et. al. [20] proposed a method using a three channel deep convolutional neural network for human activity recognition. Depth maps are first transformed into 3 WHDMMs (Weighted Heirarchical Depth Motion Maps). These WHDMMs are then fed into the three CNNs the output of which are fuesd to get final classification.

In 2010, Microsoft introduced the kinect sensor to the consumer market [11]. It was marketed as an input device for the Microsoft Xbox 360 video game console. Although initially meant as a game input device, the kinect has now become popular in research field as a cheap and easily accessible depth camera. In addition to depth images, the kinect is also capable of estimating human poses. Unlike traditional optical motion capture systems, the kinect has a significantly low barrier of entry. The kinect also outputs body joint positions in real time which makes it ideal for interactive application of activity recognition. In the past decade, a substantial amount of data have been collected by researchers for benchmark purposes from these skeleton data output of the kinect sensor. Chief among these dataset are NTU RGB+D 120 and NTU RGB+D 60 dataset[15].For a comprehensive overview of the usage of skeletal data in human activity recognition, the reader might refer to [14] and [19].The following is a overview of the most relevant literature from the field.

Quite a few researchers have explored the usage of skeletal data obtained from kinect for activity recognition. Yang and Tian [23] introduced a new descriptor called ‘EigenJoints’ based on the position differences of joints. A Naive Bayes Nearest Neighbour classifier could recognize actions from as few as 15-20 frames. Vemulapalli et al. [18] presented a different skeletal representation using the 3D geometric relationships between body parts rather then the common of representation of considering the skeleton as a set of joint positions. A different approach as adopted by Koniusz et al. [6] where they introduced a sequence kernel and a dynamics compatibility kernel to capture the higher-order statistics of how various joints related to one another in some action sequence.

Similar to depth image based activity recognition techniques, deep learning can also be applied to skeletal data to extract features. Qiuhong et al [5] generated frames from skeletal data clips and fed it into a CNN. The output of the CNN were feature vectors which a Multi-Task Learning Network used to classify actions. Shahroudy et al. [15] introduced a large dataset and proposed training a part aware LSTM to recognize activity. Their proposed P-LSTM network learns the temporal patterns of the body joints and and combines them. Yan et al. [22] argued that the reason body part aware methods work well is because they create a hierarchical representation of the skeleton. Inspired by how CNNs work on images, they proposed using a hierarchical representation of the skeleton in the form of a spatial temporal graph. Their proposed spatial-temporal graph convolution network outperformed previous state of the art techniques. SkeleMotion by Caetano et al. [3] computes the magnitude and orientation of joints to prepare a skeleton image. This is image used as the input of a CNN. Their method achieved state-of-the-art on the NTU RGB+D 120 dataset. Shi et al. [16] on the other hand combined both joint and bone information together by representing the skeleton as a directed graph network (DGN). The DGN blocks of the network could process spatial information of a single frame. In order to model the temporal aspect, they incorporate a pseudo-3D CNN along the temporal dimension. Their method achieved state-of-the-art performance on the

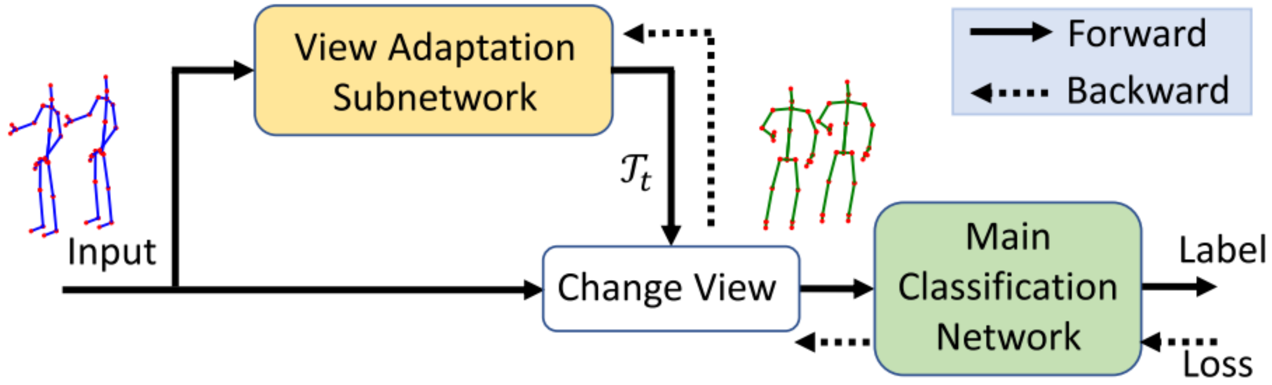

Figure 2. View adaptive Network overview[24]. τ_t is the learnable parameters of the view adaptation subnetwork.

NTU RGB+D 60 dataset.

Another approach adopted by some researchers was to convert the temporal data regarding the joint positions to a 3 channel color image. An CNN based image classification network could then identify the different classes. Du et al. [4] used this technique to get excellent result on the Berkeley MHAD dataset.

Li et al. [9] modified the technique to make it translation and scale invariant. Zhang et al. [24] farther improved the image classification based approach by adding a view adaptation sub-network. This sub-network determines the viewpoint parameters of the input skeleton sequence and transforms it so that all images are are considered from the same viewpoint for learning.

In a comparative review paper, Wang et al. [20] compared various depth and skeleton based algorithms and found that skeleton-based algorithms are more robust than depth images. Also, when given a small dataset,they found hand crafted features to out perform deep learning features. Finally, in a survey by Ren et al. [14], they reported that the accuracy of skeleton based activity recognition is already very high for the smaller NTU RGB+D 60 dataset.

3. Proposed Method

3.1. CNN based action recognition

This project was inspired by the approach presented in [4]. The key idea from that paper was as following- we can represent a sequence of 3d skeletal joint positions as a 3 channel 2d image. Each class of actions is expected to create 2d images that have some distinguishing characteristics when compared to images from other classes. A CNN based image classification network can then learn to recognize actions based on the images they create. Based on this, a neural network was implemented with the architecture as depicted in figure 1. Preprocessing the dataset so that each action is represented by an RGB image was done offline beforehand.

However, there are a few drawbacks to this. The primary one being that this approach is not view invariant. So, while the network performed similar to what was mentioned in the paper, it struggled with the more challenging NTU-RGBD 60 dataset. This is because the same action can create a significantly different image when recorded from a different viewpoint. This makes it harder for the image classification network to recognize different actions. Another issue is that the coordinates of the joints are in the coordinate space of the camera. Thus the same action performed at a different point in space will look different when transformed to an RGB image. A view-invariant approach was needed in order to solve these issues.

3.2. View-invariant approach

This problem was addressed by Zhang et al. [24] where the authors introduced a view adaptation sub-network before the classification is performed. This view adaptation part of the network is trained in an end-to-end manner alongside the classifier so that it can predict appropriate transformation parameters for the camera. The idea is as follows: Since a skeleton sequence provides the joint positions for every frame, we can represent the set of joint positions in frame t as

$$J_t = v_{1t}, ..., v_{Jt}$$

where v_{jt} is the joint position of the j'th joint at time t. This set of positions is specified in the coordinate space of the camera. However, we can easily translate the coordinate space so that the skeleton is placed in the origin at first frame of the skeleton sequence (at time t=0) through a translation transformation. This gets rid of the problem of translation variance of the skeleton sequence. To make it view-invariant, we can think of holding a virtual camera. By applying appropriate transformation to the skeleton joint positions, we can have the same effect as changing the camera view point around the skeleton. If α, β and γ are the three angles around the coordinate axis, this transformation can be represented as the composition of three rotations,

$$R = R_\alpha * R_\beta * R_\gamma$$

where R_α, R_β and R_γ are the rotation transformations. If the virtual camera is positioned at P_t then a joint v_i will be transformed as,

$$v_i' = R * (v_i - P_t)$$

The same transformation can also be applied to augment the existing dataset by some random amount to generate more training samples. Finally, a view adaptation sub-network can be trained to learn the best values of R_α, R_β, R_γ and P_t. Figure 2 depicts the approach. Thus, we have incorporated the model presented in [24] in our implementation.

3.3. View-augmentation

The same transformation already described can be applied to the existing dataset to generate variations of the samples from slightly different point of views. This can help combat the overfitting problem. So view-augmentation was also incorporated in to the implementation. The training samples are rotated randomly between [-0.3, 0.3] radian to augmented the data.

3.4. Real-time action recognition

For the action recognition network to be used for locomotion purposes in VR, it has to run in real-time. The user's skeleton sequence must be compiled into an image that can be classified by the trained network. An Azure Kinect devkit sensor was used to track the joint positions in real-time. The Azure kinect can provide skeletal joint positions in real-time with minimal delay. Every 30 frames is considered as a skeletal sequence which get converted to an RGB image before getting fed into the deep neural network. The prediction of the network can then be sent to a VR application that will make use of it for avatar locomotion.

4. Result

4.1. Dataset

Figure 3 depicts the network performance on the modified NTU-RGBD 60 dataset. All except the last class (run on the spot) have been picked from the NTU-RGBD 60 dataset. The last class (run on the spot) was taken from the larger NTU-RGBD 120 dataset. Only action classes that make some use of the lower body were selected. This was done expecting that the training data would have enough information so that the network could learn to distinguish between the lower body movements for 'run on the spot' and other actions. The 'run on spot' action was used as a substitute for 'walk in place' action due to the lack of an 'walk in place' action in the dataset. Each class had 960 samples which were split into training, validation and test sets of size 612, 32 and 316 respectively. The 'run on spot' action, being picked from a different dataset didn't match the recommended specification of the authors [15] when split by either subject ids or view ids. So it was handled manually to closely match that of the other classes.

4.2. Performance metrics

Figure 1 lists the cross subject and cross view model accuracy along with the precision and recall values for 'run on spot' class. Because of the nature of the use case of the model, the precision and recall values carry more significance than the accuracy.

Mode	Accuracy	Precision	Recall
CS	95.27	97.18	100
CV	97.13	99.38	99.38

Table 1. Cross subject and Cross view results. Precision and recall are reported only for the 'run on spot' class

5. Discussion

The network mostly performs well recognizing 'run on spot' action. It mostly faced issues when trying to distinguish between two similar actions 'put on shoe' and 'take off shoe'. However, that is not concerning for our particular use case since we care mainly about the accuracy of recognition with the 'run on spot' class.

While quite a few studies discuss the usability of walking in place as a locomotion technique, few actually provide any measure of the accuracy achieved. [8] achieved an accuracy of 99.32%. They distinguished between 'jogging-in-place' as intentional and 'marching-in-place', and squatting as unintentional actions.

In our implementation, with the cross subject model, we see that all of the 'run on spot' actions were detected successfully. A few other actions however were also miscategorized as 'run on spot', Similarly, for the cross view model the classifier was able to recognize most of the actions successfully. Since it's most likely that the network will be used in a cross subject setup, the cross subject model was incorporated into run on spot module. One advantage of this implementation is that it automatically differentiates real walking from walk-in-place.

The primary issue with the current implementation is the latency. It needs about a second of data to accurately identify the action. This may or may not be acceptable depending on the use case. A proper user study can shed more light on this.

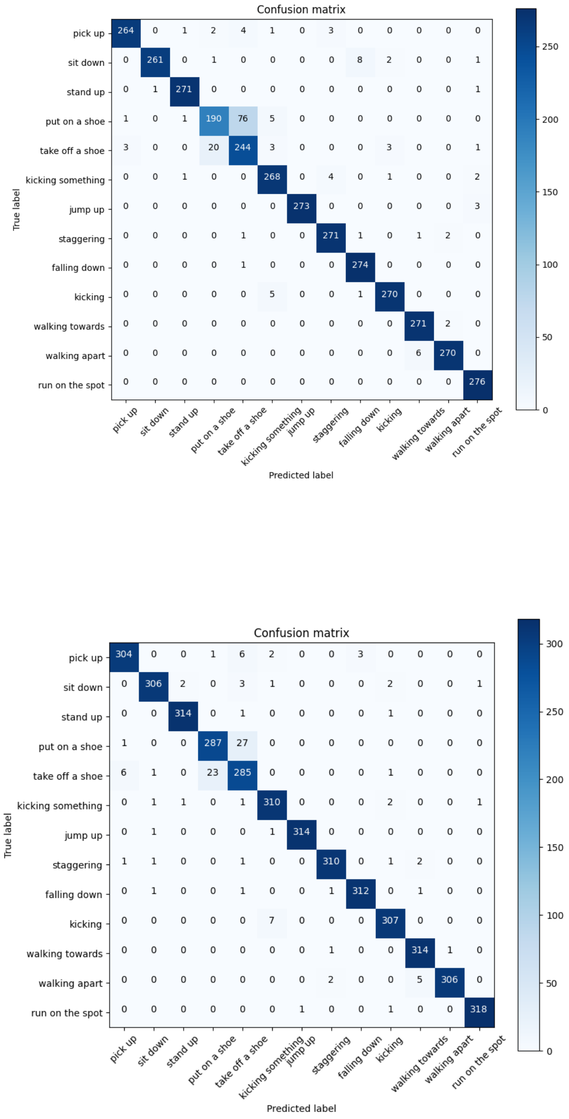

Figure 3. Confusion Matrix of accuracy on Cross Subject (top) and Cross View (bottom) dataset

Another issue is the lack of appropriate training data. For the specific use case in mind (locomotion in VR) a dataset containing actions like 'walking in place', 'running in place', 'walking' etc would be more appropriate. Such a dataset might also result in higher accuracy. Because of this, the current implementation was trained on a set of classes that seemed most appropriate intuitively.

6. Conclusion

In this paper We've presented a novel approach to recognizing walking in place action for the purpose of use in VR locomotion. The preliminary analysis shows that 'run on spot' can be successfully recognized with a high degree of accuracy. After farther improvement to the real-time evaluation speed it may be incorporated into VR applications for the purpose of locomotion.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer vision and image understanding*, 73(3):428–440, 1999.

[2] M. Al Zayer, P. MacNeilage, and E. Folmer. Virtual locomotion: a survey. *IEEE transactions on visualization and computer graphics*, 26(6):2315–2334, 2018.

[3] C. Caetano, J. Sena, F. Brémond, J. A. Dos Santos, and W. R. Schwartz. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8. IEEE, 2019.

[4] Y. Du, Y. Fu, and L. Wang. Skeleton based action recognition with convolutional neural network. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 579–583. IEEE, 2015.

[5] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3288–3297, 2017.

[6] P. Koniusz, A. Cherian, and F. Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *European conference on computer vision*, pages 37–53. Springer, 2016.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[8] J. Lee, S. C. Ahn, and J.-I. Hwang. A walking-in-place method for virtual reality using position and orientation tracking. *Sensors*, 18(9), 2018.

[9] B. Li, Y. Dai, X. Cheng, H. Chen, Y. Lin, and M. He. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 601–604. IEEE, 2017.

[10] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 9–14. IEEE, 2010.

[11] J. Lowensohn. Timeline: A look back at Kinect's history.

[12] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 716–723, 2013.

[13] H. Rahmani and A. Mian. 3d action recognition from novel viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1506–1515, 2016.

[14] B. Ren, M. Liu, R. Ding, and H. Liu. A survey on 3d skeleton-based action recognition using learning method. *arXiv preprint arXiv:2002.05907*, 2020.

[15] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[16] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019.

[17] S. Tregillus and E. Folmer. Vr-step: Walking-in-place using inertial sensing for hands free navigation in mobile vr environments. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1250–1255, 2016.

[18] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

[19] L. Wang, D. Q. Huynh, and P. Koniusz. A comparative review of recent kinect-based action recognition algorithms. *IEEE Transactions on Image Processing*, 29:15–28, 2019.

[20] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. O. Ogunbona. Action recognition from depth maps using deep convolutional neural networks. *IEEE Transactions on Human-Machine Systems*, 46(4):498–509, 2015.

[21] P. T. Wilson, K. Nguyen, A. Harris, and B. Williams. Walking in place using the microsoft kinect to explore a large ve. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 27–33, 2014.

[22] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[23] X. Yang and Y. L. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 14–19. IEEE, 2012.

[24] 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*, 41(8):1963–1978, 2019.