engineering proceedings

MDPI

Proceeding Paper

Multi-Modal Human Action Segmentation using Skeletal Video Ensembles †

James Dickens* and Pierre Payeur

School of Electrical Engineering and Computer Science, University of Ottawa, Gendron Hall, 30 Marie Curie, Ottawa, ON K1N 5N5, Canada; ppayeur@uottawa.ca

* Correspondence: jdick088@uottawa.ca

† Presented at the 10th International Electronic Conference on Sensors and Applications (ECSA-10), 15–11 30 November 2023; Available online: https://ecsa-10.sciforum.net/.

Abstract: Beyond traditional surveillance applications, sensor-based human action recognition and segmentation responds to a growing demand in the health and safety sector. Recently, skeletal action recognition has largely been dominated by spatio-temporal graph convolutional neural networks (ST-GCN), while video-based action segmentation research successfully employs 3D convolutional neural networks (3D-CNNs), as well as vision transformers. In this paper, we argue that these two inputs are complementary, and develop an approach that achieves superior performance with a multi-modal ensemble. A multi-task GCN is developed that can predict both frame-wise actions as well as action timestamps, allowing for the use of fine-tuned video classification models to classify action segments and achieve refined predictions. Symmetrically, a multi-task video approach is presented that uses a video action segmentation model to predict framewise labels and timestamps, augmented with a skeletal action classification model. Finally, an ensemble of segmentation methods for each modality (skeletal, RGB, depth, and infrared) is formulated. Experimental results yield 87% accuracy on the PKU-MMD v2 dataset, delivering state-of-the-art performance.

Keywords: Action segmentation; skeletal action recognition; video understanding; deep learning; computer vision

Citation: Dickens, J.; Payeur, P. Multi-Modal Human Action Segmentation using Skeletal Video Ensembles. **2023**, *5*, x. https://doi.org/10.3390/xxxxx

Academic Editor(s):

Published: 15 November 2023

Copyright: © 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Automated analysis of human behavior from video data has the potential to empower modern health, safety, and surveillance applications. Real-world video data mostly comes in the form of untrimmed streams, hence the motivation for developing action segmentation algorithms, where the goal is to segment a video and classify these segments, providing a dense descriptor of the sequence. Using modern deep learning techniques, human activity understanding has undergone rapid progress, wherein two prominent streams of research and application have emerged. The first such stream is the task of skeletal action understanding, involving both classification and segmentation. Deep learning approaches use large, labelled datasets [1,2], training models on sequences of skeletal frames of a fixed topology to predict action classes. To date, the most prominent approaches for skeletal action recognition employ spatio-temporal graph convolutional networks [3–6].

The second stream is the established research area of video action understanding using RGB videos [7]. Action classification networks employing 3D-CNNs [8–10] as well as vision transformers [11], currently outperform other approaches. For the task of action segmentation, the multi-stage temporal convolutional network (MS-TCN) framework [12,13] has generated impressive results, in which frame-wise features are generated in an

offline process by a 3D-CNN's backbone, after which each frame is classified with a temporal convolutional network, making use of dilated convolutions and a refinement stage. Further improvements to the MS-TCN framework can be achieved using timestamp prediction with the action segmentation refinement framework (ASRF) model [14], addressing the over-segmentation problem in which too many action segments are predicted.

In this paper, it is argued that these two streams can form complementary features for the goal of action segmentation by using a multi-modal ensemble approach. Skeletal action sequences, while sometimes noisy and dependent on the accuracy of the underlying pose estimation algorithm as well as depth map quality, are invariant to texture, lighting, clothing choice, and the layout of the scene. By contrast, while video understanding algorithms have to deal with the aforementioned factors, they allow for the incorporation of objects and surfaces within the scene to guide predictions. Consider the scenario of an individual interacting with an object, absent in a skeletal sequence, but present in a video. The presence of the object cannot be inferred by skeletal models, only the types of movements associated with the person interacting with said object.

To this end, multi-modal action segmentation approaches are developed in this work to boost segmentation accuracy. The novel contributions of this paper are the following:

- A skeletal segmentation approach that predicts both frame-wise labels and timestamps, using a video classification model to refine predictions.
- A video segmentation approach which predicts both frame-wise labels and timestamps, using a skeletal action recognition model to refine predictions.
- An ensemble of video and skeletal models, each employing their own timestamp-based refinements, to predict frame-wise labels.

Experimental results on the PKU-MMD v2 [2] dataset validate the effectiveness of these multi-modal ensembles, and state-of-the-art results are achieved using a skeletal-video action segmentation ensemble.

2. Methods

In this section the three proposed approaches are introduced, respectively a skeletal segmentation and video classification ensemble, a video segmentation and skeletal classification ensemble, as well as a skeletal-video segmentation ensemble. To maximize performance, the skeletal predictions in the approaches presented consist of a weighted ensemble of predictions made from joint, bone, joint motion and bone motion inputs as described by Shi et al. [4] and employed in state-of-the art approaches [6]. Further, video models consist of ensembles of 3 models trained on RGB, infrared (IR) and depth inputs. All ensemble weights are obtained through experimentation. An overview of the first two approaches is given in Figure 1.

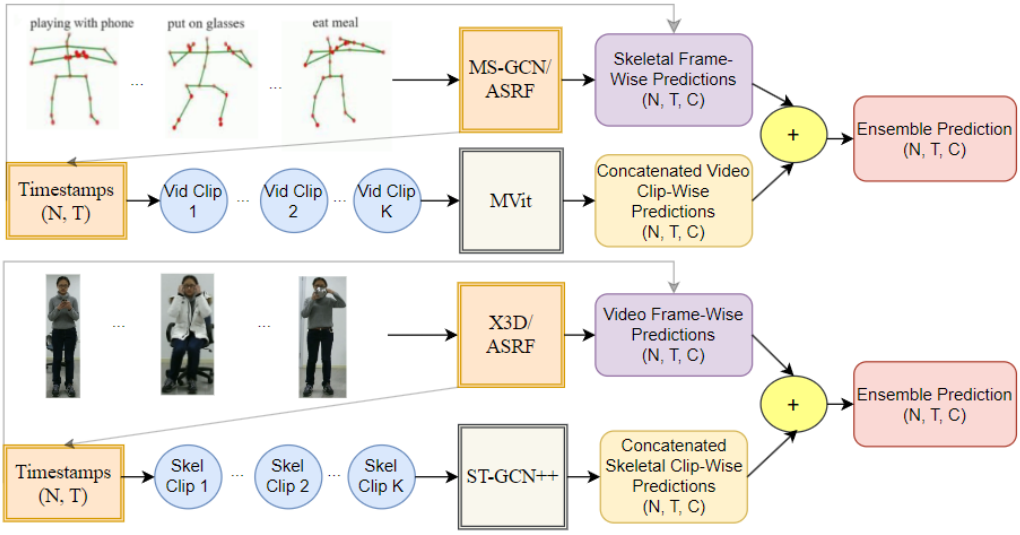

Figure 1. An overview of the proposed methods from Sections 2.1 and 2.2. Top, the skeletal segmentation and video classification ensemble. Bottom, the skeletal classification and video segmentation ensemble.

2.1. Skeletal Segmentation and Video Classification Ensemble

The skeletal segmentation backbone chosen follows the multi-stage spatial-temporal graph convolutional network (MS-GCN) [15], developed by Filtjens et al. It consists of 10 layers of spatio-temporal graph convolutions [3], where the temporal convolutions employ increasingly large dilation factors, originally set to powers of 2^i for i=1, … 9, but chosen as non-strictically increasing values determined through experimentation in this work.

The network head follows ASRF proposed by Ishikawa et al. [14]. The output of the backbone is pooled along the spatial/vertex axis and fed into a softmax layer to obtain a tensor of shape (N, T, C),

where N is the batch size, T is the number of frames, and C is the number of classes. Next, a series of refinement layers is enacted. Each refinement layer consists of a dimensionality reduction using a 1x1 convolution, followed by a sequence of dilated temporal convolutions of increasing factor (the same as the backbone), followed by a 1x1 convolution with C channels that is then input to a SoftMax layer. Note that the input of each refinement layer is the logits obtained from the previous layer. During training each refinement stage's output contributes to the segmentation loss, while during inference the output of the last refinement layer yields the model prediction.

A second branch is added that predicts action boundaries, or rather timestamps, a tensor of shape (N, T), where a value larger than 0.5 indicates the presence of an action boundary. The skeletal segmentation predictions can be refined by using a weighted sum of a majority vote of predictions within timestamps with the original frame-wise predictions. In this work we augment the ASRF framework to be multimodal. Using the predicted timestamps, the corresponding frames in the RGB, depth and IR videos are interpolated to a fixed length and fed to a video classifier to obtain class predictions, which are then expanded to the length of the clip. A weighted sum of predictions yields the final prediction. The loss for training the augmented MS-GCN model is given by:

$$L(p, gt) = \sum_{n,i} CE_{seg}\left(p_{n,i}^{seg}, g_{n,i}^{seg}\right) + \lambda_1 \sum_{n,i} CE_{ts}\left(p_{n,i}^{ts}, g_{n,i}^{ts}\right) + \lambda_2 S(p) \quad (1)$$

where p is the model prediction, gt is the ground truth, CE_{seg} is the cross-entropy loss applied framewise, n indexes over batch elements, and i indexes over refinement layers. CE_{ts} is a weighted binary cross entropy loss applied to timestamp predictions. S is the

smoothing loss given by Abu Farha et al. [12] which is used to prevent over-segmentation by enforcing consistent predictions across local windows of time, and the parameters λ_1, λ_2 weight the timestamp loss and smoothing penalty. The loss is averaged over batch size and refinement layers. During inference, non-maximal suppression with a window size of 5 is applied to timestamps to prevent near-duplicate boundaries.

The video model chosen for classification is the multi-scale vision transformer MVit [11], given its strong performance on the Kinetics-400 dataset [7]. Pre-trained weights are obtained from the PyTorch video repository (https://pytorchvideo.readthedocs.io/en/latest/model_zoo.html), obtained from training on Kinectics-400. Infrared and depth models are modified such that the weights of the input layer, consisting of a 3D convolution, are averaged across the three input channels to be able to input a single channel for efficiency. The MVit networks are trained on action-specific classification clips rather than long sequences exhibiting various actions, interpolated to 32 frames. Denoting y as MSGCN(x) for a skeletal input x, the overall segmentation prediction is given by:

$$\gamma_1 \, ASRF(y) + \gamma_2 Concat(MVit(c_1, \dots c_K)) \quad (2)$$

where ASRF denotes the majority vote refinement within skeletal segmentation predicted timestamp windows, and $Concat(MVit(c_1, \dots c_K))$ is the concatenation along the temporal axis of ordered window predictions from the video classification model for K windows, and γ_1, γ_2 are weights. The class labels per window are broadcast to the window.

2.2. Skeletal Classification and Video Segmentation Ensemble

For comparison, we experimented with a segmentation ensemble where roles of the skeletal network and video network are swapped from Section 2.1. Following standard protocol [12], video features are computed in an offline process by a 3D-CNN. We employ X3D-large [10] without a network head, using crops of 16 frames of spatial resolution (312, 312), where backbone features are pooled and reshaped to a feature vector of dimension 3888 (432 channels, 3 x 3 spatial resolution) per frame. The X3D model is pre-trained first on kinectics-400, then fine-tuned on classification training clips specific to the PKU-MMD v2 dataset split for which features were generated. The network head is the same as the skeletal model in Section 2.1, predicting both frame-wise labels and timestamps. The skeletal classification selected was the ST-GCN++ model developed by Duan et al. [6], which innovates relative to the original ST-GCN by employing a set of parallel convolutions of different kernel sizes and dilation factors in the temporal layer of the graph convolutions, concatenating the result to get more discriminative temporal features. The same loss is used as described by equation (1), where the predictions at inference for X3D frame-wise features y is given by:

$$\gamma_1 \, ASRF(y) + \gamma_2 Concat(STGCN(c_1, \dots c_K)) \quad (3)$$

2.3. Skeletal and Video Segmentation Ensemble

An alternative skeletal and video ensemble is proposed as a weighted sum of predictions from the single modality segmentation methods given in Sections 2.1 and 2.2, i.e., pure skeletal segmentation and pure video segmentation models. Since both models predict timestamps, two separate types of window-based refinement were experimented with in early trials. With a *single* refinement stage, each modality's timestamp weighted predictions are summed before applying non-maximal suppression, after which refinement occurs. With a *double* refinement scheme, each modality's predicted timestamps are used in separate specific refinements, after which the weighted sum of overall predictions is used as the final network output. Early experiments showed that the double-refinement scheme performed the best by a slight margin, and therefore is the approach considered for experiments in this work.

3. Experimental Validation

3.1. Experiments

All experiments were conducted on the PKU-MMD v2 dataset [2], which provides sequences labelled framewise for 42 different action classes in four different modalities: skeletal keypoints for single or two person actions obtained from a Kinect v2 depth camera, RGB videos of resolution 1920 × 1080, and depth and infrared maps of resolution 512 × 424. There are 13 subjects and three different camera views, where video lengths are between 1 to 2 min long with an average of 7 action instances per video. There are two splits used for testing, the cross subject (cs) in which different subjects are used for testing and training, and the cross view (cv) setup, in which different camera views are used for training versus testing. The cs split has a 773/233 train/test split, while the cv split has a 669/337 split. Due to the small number of training samples, we augmented the training set with examples from the PKU-MMD v1 dataset for the classes that overlap with v2. During training, frames with labels of 0 (the unknown class) were removed for both skeletal and video datatypes, while during inference the predictions for these frames were ignored, as is standard practice in semantic segmentation. For all networks trained, the optimizer chosen was SGD with Nesterov momentum, weight decay of 0.0005, and a cosine annealing learning rate scheduler, with all code implemented in PyTorch and trained on a single NVidia RTX 4090 GPU.

For both skeletal segmentation and classification, data augmentation consisted of light rotations about a random axis and small scaling [6]. The batch size for skeletal segmentation was 4, trained for 300 epochs, with an initial learning rate of 0.1. The batch size for skeletal classification was 16, with random uniform sampling employed [6] to yield a clip of fixed length of 150, trained for 100 epochs, using an initial learning rate of 0.01. During inference, fixed uniform sampling was used.

With respect to video segmentation, no data augmentation was used, where a batch size of 4 was used, trained for 250 epochs with an initial learning rate of 0.001. For video classification, clips were resized in an offline process to (224, 224) resolution using bicubic interpolation with anti-aliasing, after a center-wise crop. Depth and IR videos were converted to grayscale videos using min/max normalization. Random right/left flipping was used for all video modalities, and light color jittering applied to entire videos randomly during RGB video training. During training and inference, videos were interpolated along the temporal axis to 32 frames. A batch size of 6 was used, trained for 30 epochs, with an initial learning rate of 0.001.

3.2. Results and Discussion

The results of all models proposed, including skeletal and video segmentation baselines are shown in Tables 1 and 2 for the cross subject and cross view splits respectively. On both splits, the skeletal and video segmentation model exhibits the best tradeoff between accuracy and F1@50 measures, although skeletal action segmentation exhibits more of an over segmentation issue, affecting the F1@50 score negatively, presumably due to noise in the skeletons causing sudden spurious predictions. The video segmentation models struggled to predict timestamps as successfully and so the corresponding boost of adding a skeletal classifier was not successful on the cv split, and marginal on the cs split. In both splits, the video classification added on top of a skeletal segmentation showed modest improvements in accuracy and F1@50 score. In Table 3, a comparison to the MS-GCN model originally presented by Filtgens et al. [15] is shown. Note that in this original formulation, only the joint representation is used as input, hence a large performance increase is achieved using the 4 stream representation (joints, bones, joint motion, bone motion), before additionally considering the benefit of video predictions. The overall performance increase is quite large at 18.5% increase in accuracy, and 17.1% increase in F1@50 score.

4. Conclusion

In conclusion, in this work we have presented a novel approach to multi-modal human action segmentation, with experimentally validated results showing improved accuracy on the task of action segmentation using both skeletal inputs as well as video inputs (RGB, depth and infrared). As a potential avenue for future work, the use of lightweight 3D-CNNs as well as compact skeletal graph convolutional networks could be explored for the goal of real-time, or near real-time multi-modal action segmentation in order to facilitate real-world applications that are time sensitive.

Table 1. Experimental results on the PKU-MMD v2 cross subject split.

Model	Framewise Acc (%)	Segment F1@50	Timestamp–F1
MS-GCN/ASRF only (Skeletal Segmentation)	84.0	60.7	77.7
X3D/ASRF only (Video Segmentation)	82.2	73.1	64.1
MS-GCN/ASRF + MVit (Skeletal Segmentation + Video Classification)	84.6	60.7	77.7
X3D/ASRF + ST-GCN++ (Video Segmentation + Skeletal Classification)	82.6	**74.4**	64.1
MS-GCN/ASRF + X3D/ASRF (Skeletal Segmentation + Video Segmentation)	**87.0**	68.7	**79.0**

Table 2. Experimental results on the PKU-MMD v2 cross view split.

Model	Framewise Acc (%)	Segment F1@50	Timestamp–F1
MS-GCN/ASRF only (Skeletal Segmentation)	88.4	68.8	77.4
X3D/ASRF only (Video Segmentation)	76.4	64.4	61.9
MS-GCN/ASRF + MVit (Skeletal Segmentation + Video Classification)	**89.2**	70.1	77.4
X3D/ASRF + ST-GCN++ (Video Segmentation + Skeletal Classification)	75.1	65.8	61.9
MS-GCN/ASRF + X3D/ASRF (Skeletal Segmentation + Video Segmentation)	88.4	**74.5**	**79.5**

Table 3. Comparing the best performing model to other work from the literature.

Model	Framewise Acc (%)	Segment F1@50
MS-GCN [15]	68.5	51.6
X3D/ASRF + MS-GCN/ASRF (Skeletal Segmentation + Video Segmentation)	**87.0**	**68.7**

Author Contributions: Conceptualization, J.D. and P.P.; methodology, software, validation, formal analysis, J.D.; investigation, J.D., P.P.; resources, P.P.; writing—original draft preparation, J.D.; writing—review and editing, P.P.; supervision, project administration, funding acquisition, P.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by MITACS Accelerate program and NSERC Discovery grants.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.-Y.; Kot, A. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. TPAMI, 2019.
2. Liu, C.; Hu, Y.; Li, Y.; Song, S.; Liu, J. PKU-MMD: A Large Scale Benchmark for Continuous Multi-Modal Human Action Understanding. arXiv: 1703.07475, 2017.
3. Yan, S.; Xiong, Y.; Lin, D. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. AAAI, 2018.
4. Shi, L.; Zhang, Y.; Chen, J.; Lu, H. Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks. CVPR, 2019, pp. 12026–12035.
5. Chen, Y.; Zhang, Z.; Yuan, C.; Li, B.; Deng, Y.; Hu, W. Channel-wise topology refinement graph convolution for skeleton-based action recognition. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13359—13368.
6. Duan, H.; Wang, J.; Chen, K.; Lin, D. PYSKL: Towards Good Practices for Skeletal Action Recognition. Proceedings of the 30th ACM International conference on Multimedia, 2022, pp. 7351-7354.
7. Carreria, J.; Zisserman, A. Quo Vadis, Action Recognition? A new Model and the Kinetic Dataset. CVPR, 2017, pp.4724-4733.
8. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. ICCV, 2015, pp. 4489-4497.
9. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. CVPR, 2018.
10. Feichtonhofer, C. X3D: Progressive Network Expansion for Efficient Video Recognition. CVPR, 2020.
11. Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; Feichtenhofer, C.; Multiscale vision transformers. ICCV, 2021.
12. Abu Farha, Y.; Gall, J. MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation. CVPR, 2019.
13. Li, S.; Abu Farha, Y.; Liu, Y.; Cheng, M.M.; Gall, J.; MS-TCN++ Multi-Stage Temporal Convolutional Network for Action Segmentation. TPAMI, 2020.
14. Ishikawa, Y.; Kasai, S.; Aoki, Y.; Kataoka, H. Alleviating Over-segmentation Errors by Detection Action Boundaries. WACV, 2021, 2321-2330.
15. Filtjens, B.; Vanrumste, B.; Slaets, P. Skeleton-Based Action Segmentation with Multi-Stage Spatial-Temporal Graph Convolutional Neural Networks. arXiv: 2202.01727, 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.