Class-Agnostic Action Repetition Counting: Beyond Single Sequences (Progress Update)

A. Abstract

We introduce a deep learning approach to track multiple, non-overlapping repeated action sequences in dance videos. It addresses challenges in understanding repeated actions and their variations due to factors like speed, duration, and environmental conditions. We build on the architecture of Google's RepNet, a transformer-based model capable of class-agnostic action repetition counting, by adding a multilayer perceptron model called Sequence Segmenter Model (SSM) to identify boundary video frames between different action sequences. We then train this new model on an synthetic dataset that we create by concatenating videos from UCF101, a large scale action recognition dataset. We then propose an evaluation dataset consisting of 100 dance videos to assess the performance of this model. To understand the effectiveness of our SSM addition, we also a propose an approach to assess the performance of a pretrained RepNet model available from the original paper and a RepNet model that we re-train on our synthetic UCF101 dataset on the same evaluation dataset. We consider metrics such as precision, recall, F1 score, mean absolute errors, allowing us to investigate whether our modification of the RepNet architecture enhances action recognition in the dance video domain.

B. Introduction

Repeated, periodic actions are frequently observed in real world environments [2, 4]. Understanding repeated actions improves scene comprehension by facilitating the recognition of inherent patterns and building the structural coherence of a scene. For example, a person may repeat an action if an underlying motive or driving force is behind them. A challenge with classifying repeated actions is that they can vary in presentation, occurring at various speeds and durations. Lighting conditions, background noise, location, and imperfect repetitions further hinder classification. Dance is a domain where these challenges apply because it is an activity that is not restricted to a specific setting. Dance moves also vary in difficulty and intensity, which can prompt different speeds, durations, and even imperfect repetitions depending on a dancer's ability.

Tracking multiple, non-overlapping, repeated action sequences in dance videos is relevant for several reasons. Firstly, in dance education and training, precise tracking of move sequences enables instructors to break down complex choreography into manageable segments, allowing students to master each sequence with clarity and efficiency. By meticulously tracking these sequences, dancers can also analyze their performances, identify areas for improvement, and strive for greater synchronization and expression. Secondly, in the context of choreographic analysis, detailed tracking allows choreographers to identify recurring motifs and patterns, potentially informing experimentation with new combinations of movements. Finally, accurate tracking of repeated action sequences facilitates the creation of visually captivating content for dance videos, which have become increasingly popular forms of entertainment in the age of social media. It ensures seamless transitions between movements, enhances the overall flow of the choreography, and elevates the viewer's engagement and enjoyment. By tracking multiple, non-overlapping repeated action sequences in dance choreography, we may enhance dance's artistic and aesthetic quality and deepen our understanding of human movement and expression.

C. Related Work

Several machine learning models have achieved repeated action detection in specific domains. For example, the One Glimpse Early ASD detection (O-GAD) network is a temporal convolutional network that uses pyramid features of different semantic levels to construct temporal feature maps, allowing repetitive behavior detection in autism spectrum disorder (ASD) video surveillance footage [9].

Google's RepNet transcends these limitations of specialized domains by offering class-agnostic characterization of repeated actions in videos [3, 6, 10]. RepNet is a transformer-based model that identifies and counts repeated actions in videos that constitute single periods of the actions. It also constructs and uses a self-similarity matrix, allowing it to classify each video frame as belonging to a repeating temporal pattern or not. However, RepNet does not accurately count the number of repetitions in videos containing more than one sequence of repeated actions, including dance videos.

Therefore, we have two aims for building on RepNet. Our first experiment (Experiment 1) aims to develop a deep learning model capable of characterizing videos with multiple, non-overlapping repeated action sequences (RASs). The aim of our second experiment (Experiment 2) is to assess RepNet's limitations in characterizing videos with non-overlapping RASs. We also contribute two new synthetic datasets for this task designed to help our model better understand RASs.

D. Methodology

D.1. Approach

We frame the problem of keeping separate running counts for each repeated-action sequence using the same strategy that RepNet does: outputting a per-frame metric such that the set of all metrics can be used in a computation to infer a property of the entire video. Framing our problem in a manner similar to RepNet's problem statement allows us to leverage RepNet's existing architecture, which has proven to be successful at providing per-frame metrics that can be used to characterize an entire repeated action sequence.

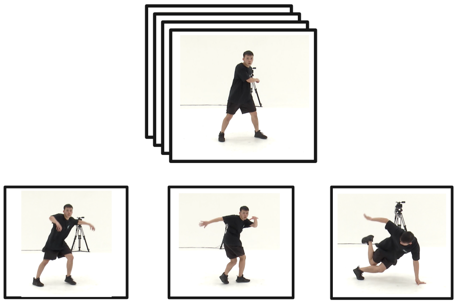

Figure 1. Example solo dance video that features multiple, non-overlapping repeated action sequences (RASs). Google's RepNet has difficulties detecting and tracking RASs in such videos.

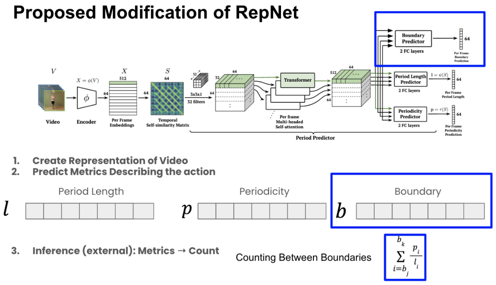
Proposed Modification of RepNet
Boundary Predictor
2 FC layers
64
Per Frame Boundary Prediction
V
$X = \phi(V)$
X
512
S
64
32
3x3x1
32 filters
Transformer
Period Length Predictor
$l = \psi(S)$
Per Frame Period Length
Periodicity Predictor
$p = \tau(S)$
Per Frame Periodicity Prediction
Video
Encoder
Per Frame Embeddings
Temporal Self-similarity Matrix
Per frame Multi-headed Self-attention
Period Predictor
1. Create Representation of Video
2. Predict Metrics Describing the action
Period Length
Periodicity
Boundary
l
p
b
3. Inference (external): Metrics → Count
Counting Between Boundaries
$\sum_{i=b_j}^{b_k} \frac{p_i}{l_i}$

Figure 2. Addition of multilayer perceptron network to Google's RepNet. This modification may address difficulties tracking multiple, non-overlapping repeated action sequences.

Currently, RepNet supports two operations (counting the number of repetitions in a repeated-action sequence and identifying the frames belonging to a repeated-action sequence's period) by providing several metrics for each frame that are used together in a post-processing computation to calculate the number of repetitions that occur in the sequence.

The two metrics that RepNet outputs for each frame are 1) the probability p_i of that frame being part of a periodic sequence and 2) the period-length l_i of the sequence to which the frame belongs. To compute the number of repetitions that occur in the video, the following computation is performed: for the ith frame, add that frame's per-frame count $c_i = \frac{1}{l_i}$ if $p_i > T$ where T is a hand-picked threshold.

To solve our problem, we propose the generation of another per-frame metric that can be used in post-processing the model output to infer separate counts for each action sequence occurring in the video.

Our proposed metric is the probability b_i of each frame being a boundary between different repeating action sequences. This new metric can be used to keep separate counts for each action sequence in the video by performing the original computation used by RepNet for frames lying between sequence boundaries: Consider all frames with $b_i > H$ for some hand-picked threshold H to be a boundary between two separate repeated-action sequences. For for each pair of consecutive boundary frames $n, m \in [0, N]$, the number of repetitions for the sequence existing between those frames is $\sum_{i=n}^{m} \frac{1}{l_i}$ if $p_i > T$.

D.2. Architecture

We propose the addition of a multilayer perceptron network, which we term the Sequence Segmenter Model (SSM), to the RepNet architecture that is responsible for predicting which frames are boundaries between repeated action sequences. The SSM is a feed-forward neural network module consisting of fully connected neurons with nonlinear activation that is compatible with video analysis [5, 7, 8].

The SSM shares the same input as RepNet's final fully connected layers for 2 reasons: (1) We hypothesize that the SSM will need an understanding of entire repeated action sequence in order to accurately predict sequence boundaries (2) We anticipate that the SSM will benefit from sharing the class-agnostic representations learned by the RepNet architecture. Thus our model also uses the same shapes as RepNet for all inputs and representations.

Thus, the architecture can be described as follows:

As per the RepNet architecture, a video $V = [v_1, v_2, ..., v_N]$ of N frames is fed into an image encoder ϕ as $X = \phi(V)$ to produce per-frame embeddings $X = [x_1, x_2, ..., x_N]$. The resulting embeddings X are used to obtain a self-similarity matrix S by computing pairwise similarities S_{ij} between all pairs of embeddings. Finally, S is fed to a module of the architecture which outputs three elements. The first element is what our paper contributes: A vector $b = \pi(S)$, produced by the SSM π where the ith component indicates if the classification of frame i is a "boundary" of a periodic sequence (that is, if a periodic sequence starts or ends on that frame the video begins to depict another periodic or aperiodic sequence).

The second and third elements outputted by the model were present in the original RepNet Architecture: $l = \psi(S)$ containing in the ith component a period-length estimate, periodicity a boundary classification ”score $p = (S)$.

D.3. Dataset

We are unaware of an existing dataset with videos containing multiple repeat action sequences and thus aim to automatically synthetically generate the dataset. Automatically synthesizing these videos saves time and manual labor, and we anticipate that training on synthesized videos will come at no cost to performance since the RepNet architecture, which is being used in our new model, is able to successfully generalize to real videos even when trained solely on synthetically generated videos of repeated actions.

We synthesize realistic videos that possess periodic motion sequences using the following process: The creators of RepNet [3] proposed the use of a pipeline $f()$ for producing, from any given video W, a video $W' = f(W; n, l)$

that exhibits a length $l = |v|$ subsequence $v \subset W'$ containing periodic motion with n repetitions, and smooth transitions between periodic (v) and aperiodic $(W' \setminus v)$ portions of the video $f(W; n, l)$. This technique allowed the authors to quickly gather an annotated realistic dataset for the task of characterizing periodic motion that was large enough to train on. The dataset proved to be realistic enough for the RepNet architecture, being soley trained on this dataset, to outperform benchmarks on real datasets. Thus we take their approach to create videos with several repeated action sequences: We select a random partition of a given video V into 2 sequences $V_1 V_2 = V$ and concatenate to create a video $V' = f(V_1; n_1, l_1) f(V_2, n_2, l_2)$ that contains $\nu = 2$ distinct repeated action sequences. We randomly select each subsequnce's repetition count n_i from the set of integer factors of l_i that are greater than 2. And l_i is randomly selected from the set of integers between 3 and $|V_i|$.

Since RepNet architecture has been shown to train optimally when the videos have frame-length of 64, our dataset consists of frame-length 64 videos.

The number of repeated action sequences occurring in a single video ν is a hyperparameter of our method that must be optimized through experimentation. When producing our synthetic video V' from a source video V, we currently select no greater than 2 separate sequences in the same video due to the fact that allowing allowing too many repeated sequences to exist in a fixed-length (64 frames) video would cause our pipeline to produce videos with period lengths that are too small to be realistic or common.

The source of our videos, UCF101 dataset, features 13320 YouTube videos from 101 action categories. These action categories can be divided into the following: semantic settings, human-object interaction, human-human interaction, playing musical instruments, sports, and body-motion only. The videos included in this dataset also exhibit variations in camera motion, viewpoint, illumination, object scale, and object appearance and pose, making our synthetic dataset representative of a large number of applications.

For our evaluation dataset, we use the AIST Dance Video Dataset. This publicly available dataset features 13940 videos, including dances from ten major genres, solo and group dances, dances of different difficulties, and different camera angles and viewpoints, and different dance contexts (e.g. dance showcase versus dance battle). Therefore, we randomly sample without replacement 100 videos from this dataset to create our evaluation dataset.

D.4. Training

We train each model with a learning rate of $6 * 10^{-}6$, 400K steps, batch size of 5, and an Adam optimizer for 50 epochs. For updating model parameters, we use two different loss functions. We use binary cross entropy loss to optimize our model's ability to determine boundary and periodicity of the multiple repeated actions that occur in our "extended" data. Both features represent binary classification outcomes for each frame in the video. We also use a multi-class classification objective to optimize its ability to detect period length. WE chose categorical cross-entropy because period lengths for each frame were represented as one-hot encodings to be compatible with predictions made by original RepNet model.

D.5. Experiments

Similar to the original RepNet paper, we implement our different experiments with Tensorflow [1]. For our first experiment, we evaluate the performance of the pre-trained RepNet model, which Google Research made publicly available, on our evaluation dataset consisting of dance videos. This model was trained on Countix, a dataset featuring 8757 YouTube videos of single repeated action sequences in in various semantic settings, such as workout activities, artistic performances, and sports.

For our second experiment, we re-initialize RepNet's weights while preserving its architecture. We then train this model on our synthetic UCF101 dataset. Unlike Countix, our synthetic UCF101 dataset features videos of multiple repeated action sequences.

For our final experiment, we build on the RepNet architecture by adding a multilayer perceptron network called Sequence Segmenter Model (SSM). This SSM is responsible for predicting which frames are boundaries between repeated action sequences. We want to see if this feature improves its detection of repeated action sequences compared to the models in our first two experiments.

D.6. Evaluation Metrics

Boundary Classification. To understand the effectiveness of the SSM that we add to the original RepNet architecture, we evaluate the boundary classification of our RepNet + SSM model. We do not consider the other two RepNet models because they are incapable of boundary classification. The SSM outputs the probability of a frame dividing two different repeated action sequences. We turn it into a per-frame binary classification task by testing several thresholds for decision boundaries. If the probability is above the threshold, we assume the RepNet + SSM model classifies the current frame as a boundary frame. If the probability is below the threshold, we assume our model classifies the current frame as a boundary frame. We define a true positive (TP) as correct classification of boundary frame and a false positive (FP) as misclassification frame as boundary frame. We also define a false negative (FN) as misclassification of a non-boundary frame and a true negative (TN) as correct classification of a boundary frame. We then use these metrics to calculate precision and recall.

Periodicity. To understand how RepNet + SSM model compares to pre-trained RepNet model (baseline) and the one trained on our synthetic UCF101 dataset, we also evaluate periodicity classification. Specifically, we consider precision and recall because periodicity classification is also a per-frame binary classification task. We define true positive (TP) as correct classification of a video frame as part of a repeated action sequence. We define false positive (FP) as misclassification of a video frame as part of a repeated action sequence. We then define false negative (FN) as misclassification of a video frame as not part of repeated action sequence and true negative (TN) as correct classification of a video frame as not part of a repeated action sequence.

Mean Absolute Error (MAE). To evaluate repetition counting, we use Mean Absolute Error (MAE) which is discussed in the original RepNet. MAE is based the absolute difference between the ground truth count of a repeated action sequence and the count that our model predicts. The absolute difference is normalized to a range of 0-1 by dividing by the ground truth count, and we can average these normalized values across the 50 videos in our dance video evaluation dataset to obtain MAE.

Temporal Self-Similarity Matrix (TSM). To qualitatively evaluate our models' representation of repeated dance moves in videos from our evaluation dataset, we also consider the temporal self-similarity matrix (TSM) used as an intermediate layer in the RepNet architecture. The Temporal Self-Similarity (TSM) represent human action recognition in deep learning model, serving as a valuable visual representation of how our different models represent dance moves in our evaluation dataset.

E. Experimental Results

E.1. Limitations

Despite the constraints of Tensorflow, we eventually implemented a training loop for our model that incorporated our outlined training parameters. This training loop incorporated learning rates, batch sizes, and optimizer configurations tailored to our specific requirements. The training loop was constructed to handle data preprocessing, model feeding, loss calculation, and backpropagation during the training process with our synthetic dataset. However, we could not implement the RepNet + SSM model due to the complexity of freezing intermediate layers within the RepNet architecture and writing code capable of statistical analysis and data transformation for RepNet to be capable of boundary classification.

We encountered challenges when attempting to integrate the RepNet + SSM model. The primary difficulty with implementing the RepNet + SSM model stemmed from the complexity of freezing specific layers within the RepNet architecture. In the context of the RepNet, the early convolutional and encoder layers are typically responsible for processing input data into a meaningful representation. Freezing these layers means that the decoder and any later modules, including the SSM, can build on a stable and reliable foundation, learning to generate outputs that are directly useful for recognizing action sequences. However, the process of determining which combinations of layers to freeze and steps to do proved quite challenging given the constraints of Tensorflow, which has various deprecated functions and commands.

Model	Recall	Precision	F1
RepNet (Countix)			
RepNet (UCF101)			
RepNet + SSM (UCF101)			

Table 1. Boundary classification results for different models

Model	Recall	Precision	F1
RepNet (Countix)			
RepNet (UCF101)			
RepNet + SSM (UCF101)			

Table 2. Periodicity classification results for different models

Adapting RepNet for a boundary classification task introduced another layer of complexity. To repurpose RepNet, we needed to modify the output layer to handle discrete boundary prediction. This task involved considering the extraction and transformation of output data from the original RepNet architecture into a suitable format. The format needed to capture the essential characteristics required for accurate prediction of boundary frames, adding to the complexity of the task.

To address these challenges, our future steps would have been further to investigate RepNet's architecture and TensorFlow's capabilities and devise a systematic approach for layer freezing. This strategy might involve experimenting with different subsets of layers to freeze and evaluate the resultant models. We would have also considered more statistical and data transformation methods for RepNet to achieve boundary classification. These enhancements would require iterative testing and validation to ensure they meet the criteria for our specific application.

Boundary and Periodicity Classification. We would have provided precision, recall, and F1 metrics in table like **Table 1 and 2**. Since periodicity and boundary classification are binary classification tasks, we would have expected the SSM to be particularly helpful addition that would increase the precision, recall, and F1 score of our model. The SSM could allow more frames to be correctly classified by allowing our model to better understanding the start and end of repeated action sequences. We would have also expected our RepNet model trained on our synthetic UCF101 dataset to exhibit higher precision, recall, and F1 score compared to the baseline pretrained RepNet model. The pretrained RepNet was trained on Countix dataset videos of single action sequences, making it less equipped to detect the start of new repetitive sequences. Conversely, our synthetic UCF101 dataset features videos with multiple repeated action sequences that train RepNet to better handle the temporal segmentation of video into action sequences.

Mean Absolute Error (MAE). We would have provided the MAE results in a table that looks like **Table 3**. We would have expected the RepNet + SSM model, which

Model	MAE
RepNet (Countix)	
RepNet (UCF101)	
RepNet + SSM (UCF101)	

Table 3. Mean Absolute Error (MAE) results for different models

were trained on the UCF101 dataset, to exhibit the lowest MAE because the SSM enhances the detection of repeated action sequence boundaries. Within the context of repetition counting for solo and group dances of varying difficulty and camera angles, additional parameters devoted to detecting both the start and end boundaries of actions improves the model's ability to recognize dance moves, potentially addressing issues like double counting. We then would have expected the RepNet model trained on our synthetic UCF101 dataset to exhibit a lower MAE than the baseline pretrained RepNet model. Similar to periodicity classification, the underlying reason is that our synthetic UCF101 dataset features videos with mulitple, non-overlapping repeated action sequences as opposed to videos with a single repeated action sequence. Therefore, the RepNet + UCF should generalize more to our evaluation dataset.

We were able to implement a training loop for our model that incorporated our outlined training parameters. However, we were unable to implement the RepNet + SSM model due to the complexity of freezing intermediate layers within the RepNet architecture and adding necessary statistic analysis code for RepNet to be capable of boundary classification task.

F. Discussion

We contribute a large annotated dataset for the task of localizing repeated action sequences within videos. Of datasets that have several repeated actions per video, this is the first dataset that exhibits a wide variety of subjects and as many as 101 action categories. This dataset can be used to train other network to perform class-agnostic localization of repeated action sequences.

We designed experiments to evaluate periodicity, repetition counting, and period length detection, using metrics such as precision, recall, F1 score, Mean Absolute Error (MAE), and categorical cross-entropy loss. And the results would show improvements in these metrics compared to the baseline RepNet model, indicating the efficacy of incorporating the SSM into the architecture.

Qualitatively, we observed improvements in the representation of repeated dance moves in videos through visual analysis of temporal self-similarity matrices (TSMs). The addition of the SSM enhanced the TSM's ability to capture the temporal relationships between frames, leading to more accurate characterization of repeated action sequences.

Our study contributes to advancing the field of action recognition and scene comprehension by addressing the challenge of characterizing videos with multiple repeated action sequences.

G. Conclusion

In this study, we introduced a deep learning approach for tracking multiple, non-overlapping repeated action sequences in dance videos, addressing the limitations of existing models like Google's RepNet. By integrating a multilayer perceptron-based Sequence Segmenter Model (SSM) into the RepNet architecture, we enhanced its ability to identify boundaries between different action sequences. Our synthetic dataset, created from UCF101, provided a robust training ground, enabling the model to handle the complex nature of dance videos. Although we faced implementation challenges that greatly limited our ability to evaluate and compare our model, our proposed methodology and experimental design could potentially improve action recognition accuracy if implemented. This idea contributes to the broader field of action recognition by offering a possible pathway for more effective analysis and segmentation of complex video sequences, particularly in dynamic and varied environments such as dance.

H. Breakdown of Work

Oreo focused on developing the code necessary to produce synthetically generated datasets from the UCF101 dataset. Additionally, she was responsible for implementing the SSM module that we planned to incorporate. David, on the other hand, handled the selection of test dataset and its annotations. He also processed the data based on Oreo's code due to his access to MIT Supercloud GPUs, created the training loop for the data, and wrote the code to calculate evaluation metrics for a specified input format

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning, 2016. 3

[2] Ashis Kumar Chanda, Chowdhury Farhan Ahmed, Md. Samiullah, and Carson K. Leung. A new framework for mining weighted periodic patterns in time series databases, 2017. 1

[3] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild, 2020. 1, 2

[4] Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. Mining periodic behaviors for moving objects, 2010. 1

[5] Fionn Murtagh. Multilayer perceptrons for classification and regression, 1991. 2

[6] Saptarshi Sinha, Alexandros Stergiou, and Dima Damen. Every shot counts: Using exemplars for repetition counting in videos, 2024. 1

[7] Jonti Talukdar and Bhavana Mehta. Human action recognition system using good features and multilayer perceptron network, 2017. 2

[8] Haiman Tian and Shu-Ching Chen. Mca-nn: Multiple correspondence analysis based neural network for disaster information detection, 2017. 2

[9] Yuan Tian, Xiongkuo Min, Guangtao Zhai, and Zhiyong Gao. Video-based early asd detection via temporal pyramid networks, 2019. 1

[10] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation, 2019. 1