frontiers in Computer Science

ORIGINAL RESEARCH
published: 07 March 2022
doi: 10.3389/fcomp.2022.806027

Exploiting the Nature of Repetitive Actions for Their Effective and Efficient Recognition

Konstantinos Bacharidis [1,2]* *and Antonis Argyros* [1,2]

[1] *Computer Science Department, University of Crete, Heraklion, Greece,* [2] *Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH), Heraklion, Greece*

In the field of human action recognition (HAR), the recognition of actions with large duration is hindered by the memorization capacity limitations of the standard probabilistic and recurrent neural network (R-NN) approaches that are used for temporal sequence modeling. The simplest remedy is to employ methods that reduce the input sequence length, by performing window sampling, pooling, or key-frame extraction. However, due to the nature of the frame selection criteria or the employed pooling operations, the majority of these approaches do not guarantee that the useful, discriminative information is preserved. In this work, we focus on the case of repetitive actions. In such actions, a discriminative, core execution motif is maintained throughout each repetition, with slight variations in execution style and duration. Additionally, scene appearance may change as a consequence of the action. We exploit those two key observations on the nature of repetitive actions to build a compact and efficient representation of long actions by maintaining the discriminative sample information and removing redundant information which is due to task repetitiveness. We show that by partitioning an input sequence based on repetition and by treating each repetition as a discrete sample, HAR models can achieve an increase of up to 4% in action recognition accuracy. Additionally, we investigate the relation between the dataset and action set attributes with this strategy and explore the conditions under which the utilization of repetitiveness for input sequence sampling, is a useful preprocessing step in HAR. Finally, we suggest deep NN design directions that enable the effective exploitation of the distinctive action-related information found in repetitiveness, and evaluate them with a simple deep architecture that follows these principles.

Keywords: deep learning, repetition localization, video understanding, human activity recognition (HAR), action recognition

OPEN ACCESS

Edited by:
Juergen Gall,
University of Bonn, Germany

Reviewed by:
Yazan Abu Farha,
University of Bonn, Germany
Sovan Biswas,
Intel, Germany

****Correspondence:***
Konstantinos Bacharidis
kbach@ics.forth.gr

Specialty section:
This article was submitted to
Computer Vision,
a section of the journal
Frontiers in Computer Science

Received: *31 October 2021*
Accepted: *31 January 2022*
Published: *07 March 2022*

Citation:
Bacharidis K and Argyros A (2022)
Exploiting the Nature of Repetitive
Actions for Their Effective and Efficient
Recognition.
Front. Comput. Sci. 4:806027.
doi: 10.3389/fcomp.2022.806027

1. INTRODUCTION

Human activity analysis and understanding in videos is an important task in the field of computer vision. Its applications range from Human-Robot Collaboration (HRC), assistive technologies for daily living, up to surveillance and entertainment. The importance of these tasks is accompanied with significant challenges, due to the high-dimensionality of video data and the changes in the appearance characteristics due to scene/context and viewpoint variations (Herath et al., 2017). These challenges become severe when discriminating among different *fine-grained* actions in action sets with high intra- and low inter-class appearance and motion similarities (Aggarwal and Ryoo, 2011).

One particular challenge that becomes more evident as we proceed to model complex and/or fine-grained activities consisting of multiple actions or action steps, is the temporal execution extent of the action. Action execution duration varies for each human. The simpler the action, the more temporally constrained is its duration, and thus, the temporal extends of the short and long-term information that a model needs to assimilate. As the complexity increases, so does the duration and the execution variations. A robust action recognition model has to be able to model both short- and long-term appearance/motion information of the action execution (Aggarwal and Ryoo, 2011; Kang and Wildes, 2016).

Robust short-term modeling has been achieved in the last decades with elaborate hand-engineered, and more recently deep learning-based, feature descriptors. Long-term modeling is still an issue in the deep learning era, since the generation of robust representations through the hierarchical correlation of deep features does not scale well as the duration of an action increases (Zhu et al., 2020). This has an additional impact on the computational cost both for training and inference. Therefore, it is important to investigate strategies that can provide a compact and discriminative representation of a long input sequence demonstrating an action, either by selecting the most informative action-specific temporal segments of the sequence or by leveraging cost-efficient and easy to compute aggregation of information along the action duration. Existing approaches orbit around sparse sampling, clip cropping or segment-wise processing and aggregation of the input sequence, favoring either short or long-term dependencies for the shake of computational efficiency.

One action/activity category that does not benefit from sparse sampling or clip cropping is *repetitive actions*. This is due to the fact that for these action cases, such approaches lead to temporal ordering disruption and/or redundant information processing. Repetitive actions are quite common, especially in daily living (e.g., cooking, physical exercise, etc.), with the core execution task being repeated with slight variations. Due to their nature, these actions contain redundant information regarding coarser appearance and motion characteristics. Moreover, in such actions, we can pinpoint the gradual effect of the repetitive task in the scene, if any. As a gradual effect we define any change that the repetitive action causes on the appearance state of the object(s) in use, of the scene or of the actor. The objective of this article, is to explore and exploit the nature of repetitive tasks as a way to (a) reduce sequence length by removing the repetitive executions, resulting in more distinct and compact representation of the action pattern and (b) highlight the gradual effects of the repetitive action in the scene and objects, allowing the model to consider them as an action-related attribute during learning. To the best of our knowledge this work is the first to study the characteristics of repetitive actions within the scope of HAR, and to propose a first pipeline that enables the effective distillation and exploitation of such information.

2. RELATED WORK

Input sequence sampling: HAR methodologies use two main strategies to perform sequence sampling, (a) randomly cropping a clip from the sequence and (b) uniformly splitting sequence into snippets and sampling a key-frame, either raw or by applying some pooling or temporal ranking operations. These techniques are applied to both hand-crafted and deep learning HAR approaches. Our analysis focuses on deep learning HAR due to its prevalence on the field.

Regardless of whether they perform random cropping or uniform splitting, existing deep learning approaches usually end-up with sampled sequences in the range of 16, 32, or 64 frames. Random cropping has strong short-term, but weak long-term information content, whereas uniform splitting is opposite in nature. In both cases, due to the partial observation of the action with these two sampling schemes, researchers designed models that are capable of highlighting and exploiting dependencies between sparse input sequences. This is achieved either with two-stream CNN models, using appearance (RGB) and motion (optical flow) inputs (Simonyan and Zisserman, 2014; Feichtenhofer et al., 2016) sometimes combined with memorization RNN cells to increase the long-range modeling capabilities (Donahue et al., 2015; Varol et al., 2017), or with the use of 3D convolutions, along with pooling operations, to directly learn spatio-temporal representations (Tran et al., 2015; Carreira and Zisserman, 2017). In complex action or activity cases, both short-term and long-term dependencies are important. Thus, increasing the portion of the input sequence to be processed becomes a necessity. Recent methods apply temporal pooling or deep encoding on snippets of the sequence to encode the short-term dependencies, and use these encoded segments as the input sequence components. Large-scale recognition is then performed in two ways, either using consensus criteria on per-snippet action estimates derived from each short-term temporal encoding (Wang et al., 2016), or working with the short-term snippet-driven feature maps and applying to them temporal convolutional operations (Bai et al., 2018; Zhang et al., 2020) or generic non-local operations (Wang et al., 2018).

Periodicity estimation and repetition counting: Repetition localization is achieved *via* the robust periodicity estimation in time series. For video sequences, periodicity detection is performed by examining the spatio-temporal feature correlations in a self-similarity assessment fashion, with the most successful strategy being to create a Temporal Self-similarity Matrix (TSM), using hand-crafted, motion-related (Panagiotakis et al., 2018) or deep learning-based (Karvounas et al., 2019; Dwibedi et al., 2020) frame-wise representations. The identification of periodicity in a sequence, allows us to count the repetitions of the task using period length predictions. Existing works on repetition counting (Levy and Wolf, 2015; Runia et al., 2018; Dwibedi et al., 2020), formulate the problem as a multi-class classification task, with each class corresponding to a different period length. Repetition counting is then performed by evaluating the entropy of the per-frame period lengths predictions (Levy and Wolf, 2015) as well as per-frame periodicity predictions

(Dwibedi et al., 2020). We built upon the work of Dwibedi et al. (2020) and utilize the counting process to localize and segment each repetition sequence.

3. REPETITIVENESS IN ACTION RECOGNITION

In repetitive actions, each repetition sequence preserves the core action motif, while deviations mainly consider the action execution tempo, and action impact on the scene and objects. This means that we can get a better understanding of the core pattern of the action and the action effects to the surrounding environment, by distinctly exploring each repetition sequence.

3.1. Characteristics of Repetitive Actions

We pinpoint three characteristics of repetitive actions that are important when trying to exploit action repetitiveness for HAR. These are (a) the *number of repetitions*, (b) the *variability of the repetitions*, and (c) the *presence/absence of action-imposed gradual effects on the surrounding scene*.

Number of repetitions: It is likely that as the number of repetitions increases, the information redundancy in the repetitive segments, also increases. Thus, in the case of few repetitions, it is more likely that it is necessary to model the entire sequence. In the limit, a single, non-repetitive action requires full modeling.

Variability of repetitions: The number of repetitions is a simple indicator of information redundancy, which, nevertheless, needs to be accompanied by a measure of the variability of the repetitive segments. Repetitions are completely redundant if they are identical, independently of their number. The larger the variability among different repetitions, the more the information content they offer and the larger the need of being modeled.

Gradual effects: Repetitive actions may (or may not) have an effect on the actor and/or the surrounding scene. For example, actions such as *clapping* do not impact the surrounding scene. Such repetitive tasks may exhibit variability (as explained before) due to, e.g., tempo change, differences in execution style, etc. On the other hand, the action of *slicing a fruit/vegetable*, has an additional gradual effect on the element/object it is applied to. Importantly, the nature of these gradual effects is quite characteristic for the action and may have strong discriminative power, especially in the disambiguation of actions that share similar motion such as *cutting in slices* and *cutting in cubes*. Based on this, to exploit repetitiveness, we need to consider (a) the *definition of the core execution motif of the activity* and (b) its *gradual effects*, i.e., the impact on the surrounding space.

3.2. Highlighting Action-Related Effects With Repetitiveness

The key advantage of sequence splitting based on repetitiveness is that it allows to decouple and highlight the gradual variations that may occur in the surrounding scene due to the effect of the performed action. In addition, it makes it easier to detect slight variations on the way the action is performed, such as tempo changes. As we previously mentioned this information can be important in actions with similar appearance/motion characteristics (low inter-class variance) which, however, differ on the gradual changes that the action imposes on the scene or on object-of-interest. For such cases our proposal is to work with the repetition sequences with the goal of highlighting the gradual action effects. We can consider the core execution pattern as the short-term action dynamics whereas the gradual changes to the object-of-interest or scene as the long-range ones. Ideally, we would like an HAR model to be able to access both information sources at the same time without any information loss, however, this is not feasible due to hardware limitations and model footprint (Zhu et al., 2020). As a solution, we propose to restrict the repetition sequence lengths by employing sequence summarization or temporal encoding and rank pooling methods, such as Dynamic Images (DIs) (Fernando et al., 2015; Bilen et al., 2017), Motion History Images (MHIs) (Ahad et al., 2012), or a deep encoder network (Wang et al., 2016, 2021). The resulting embedding encodes, in a compact way, the temporal action dynamics as well as the action impact on action-affected scene elements. For example, for the *slicing onion* case, DIs or MHIs capture the effect of the action on the onion, as shown in **Figure 1**.

Under the above considerations, for the case of actions with *no action-invoked gradual effects*, we expect that a repetition-based feature encoding would result in representations that are mapped tightly/closely in the feature space. On the other hand, for actions with *action-invoked gradual effects*, we expect the mappings to be sparser. To verify this hypothesis, we select two videos from the action classes (a) *running on treadmill*, (no gradual effects) and (b) *slicing onion* (with gradual effects). We use a simple pre-trained I3D to generate the repetition segment-based temporal encodings, resulting in a $1 \times 2,048$ feature vector per repetition segment. To visualize these representations, we applied Principal Component Analysis (PCA) Pearson (1901), Abdi and Williams (2010). **Figure 2** illustrates the feature space defined by the first three principal components. We also provide the corresponding DIs purely for visualization purposes. As it can be verified, the illustrated mappings verify the presence of information redundancy for actions with no or subtle gradual effects on the surrounding space, as well as discriminative elements among the repetition segments of actions with gradual effects.

4. REPDI-NET: A DEEP ARCHITECTURE TO EXPLOIT ACTION REPETITIVENESS

In order to model the core execution pattern and the gradual scene changes due to repetitiveness, effectively and simultaneously, we propose an HAR deep neural network architecture, dubbed *RepDI-Net*. The proposed architecture comprises of two modules. The first is a data pre-processing module, whose goal is threefold, (a) to identify a reference execution of the action, (b) to estimate a set of coefficients that express the underlying similarity between the repetition segments, and (c) to generate a sequence consisting of temporal encodings of the repetition segments [for this our model exploits the temporal rank-pooling approach of *Dynamic Images*

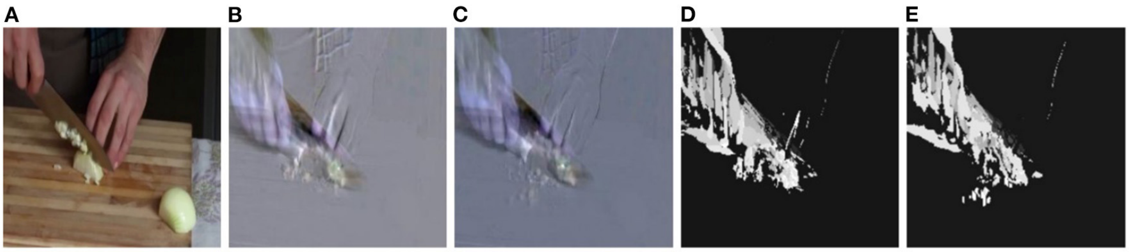

FIGURE 1 | Repetitive action effect encoding using Dynamic Images (DIs) and Motion History Images (MHIs). **(A)** Keyframe from the first repetition, **(B,C)** DIs of 1st and 2nd repetition segments, and **(D,E)** MHIs of the 1st and 2nd repetition segments.

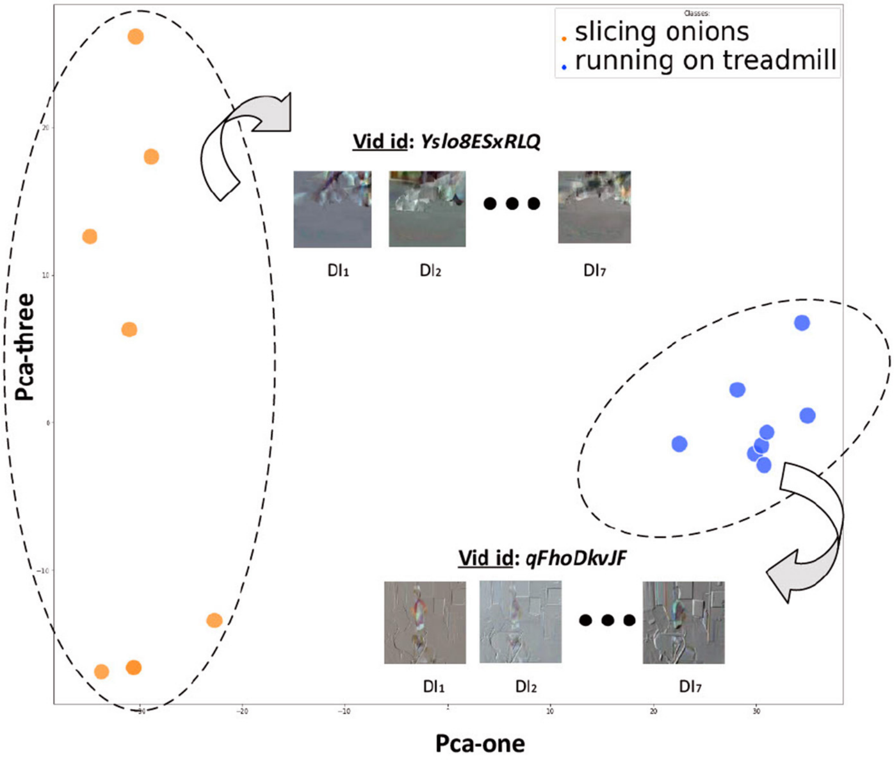

FIGURE 2 | Example of potential information redundancy due to high inter-repetition segment similarity (blue case) and presence of repetition segment-wise discriminative information due to potential action-invoked effects on the surrounding environment (orange case). Each point corresponds to a repetition segment of the specific video sample.

(DIs), (Fernando et al., 2015)]. The second module is a two branch spatio-temporal sequence modeling DNN, that utilizes the aforementioned information streams to perform the action classification task. An overview of the architecture is shown in **Figure 3**.

In this section, we examine the key elements of the data preprocessing module of *RepDI-Net*, highlighting their role and importance in the effective modeling of a repetitive action in the second module of the proposed architecture. Details regarding the specifications for the spatio-temporal sequence modeling DNN module can be found in section Experiments.

4.1. Reference RGB Execution Sequence

The input to the first sub-net of *RepDI-Net*'s spatio-temporal sequence modeling module consists of the core temporal appearance information of the action using the raw RGB sequence of a reference execution of the action (can be the first or any other repetition). The use of appearance information is important for distinguishing actions, as it provides scene-centered, texture-related information. Any execution of the task can be used as the reference appearance input source. However, we would like the reference execution sequence to be (a) free from cases of background clutter or occlusions and (b) as similar

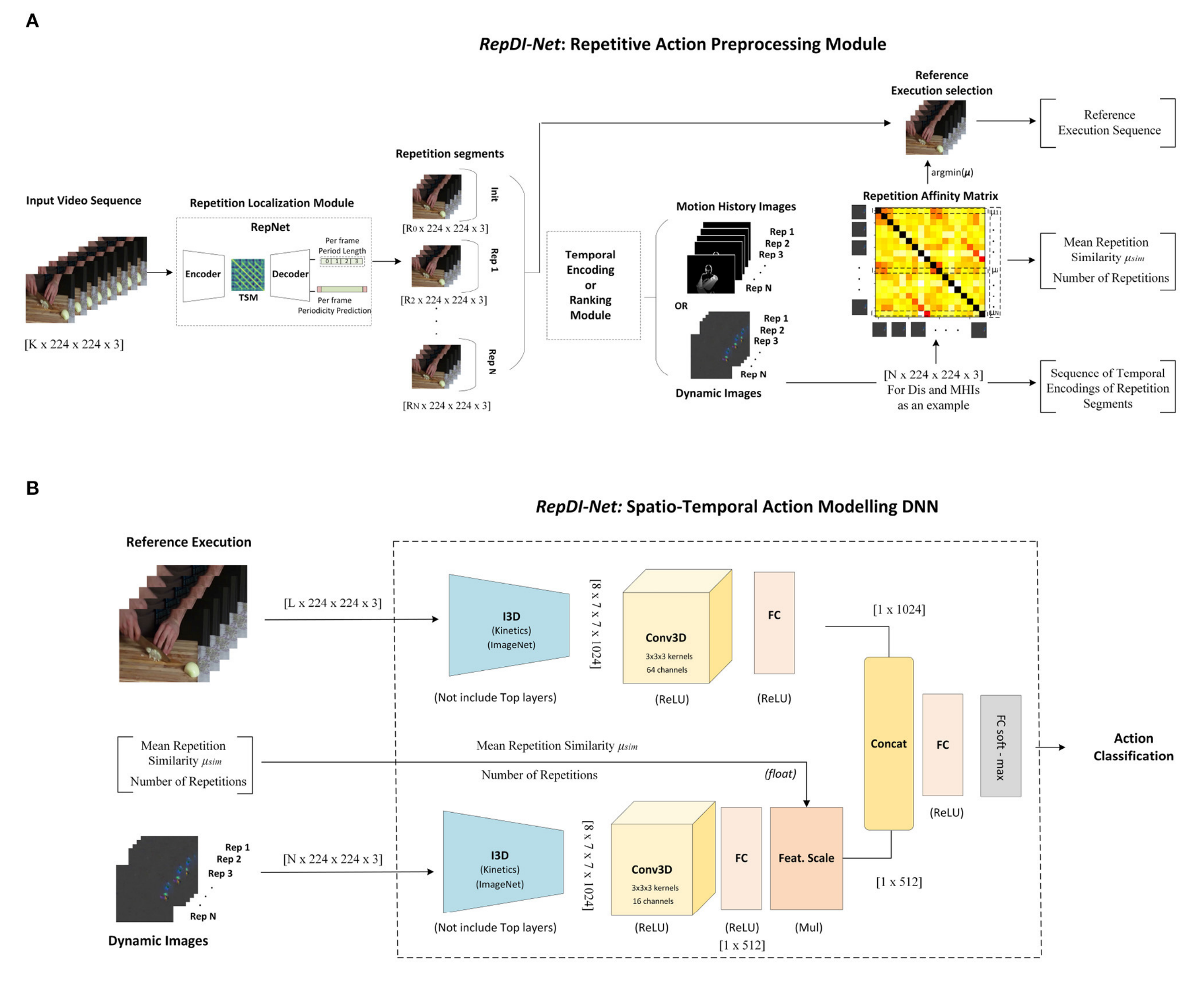

FIGURE 3 | *RepDI-Net*: a deep HAR architecture to exploit the repetitive action segments. **(A)** Input sequence preprocessing module and **(B)** Spatio-temporal sequence modeling and classification module. Regarding the temporal encoding/ranking examples, each Dynamic Image (DI) and Motion History Image (MHI) corresponds to a single repetition.

as the rest of executions (repetitions). As a way to resolve this, we construct a *Temporal Self-similarity Matrix* (TSM), which comprises of the pairwise Euclidean distances of the repetition DIs. We define as the reference execution instance the one that, on average, is most similar to all the others. Thus, we select the one that has the minimum average Euclidean distance to the rest of the instances/repetitions.

4.2. Distilling Information From Repetitions

The input of the second sub-net consists of the temporal encodings of the repetition sequences. As hinted earlier, in our experiments, we use DIs for this task due to the richer encoding attributes they possess against MHIs (Bilen et al., 2017). However, any temporal encoding or ranking approach can be employed (Wang et al., 2016; Cherian et al., 2017; Diba et al., 2017; Lin et al., 2018). The role of the second sub-net is to highlight the characteristic information that is present in the repetition sequences from the redundant information (consisting of the almost identical appearance and motion features between repetitions). To teach the model when it is useful to focus on this aspect, we include two factors as additional inputs to the second sub-net, which act as feature scaling components in the last Fully-Connected (FC) layer, f_{DI}, of the repetition sub-net:

$$f_{DI_{sc}} = \left(1 - \frac{\mu_{sim}}{N_{rep}}\right) f_{DI}. \quad (1)$$

In Equation (1), $\{f_{DI}, f_{DI_{sc}}\} \in \mathbb{R}^{1\times d}$, where d is the dimensionality of the output of the FC layer. Equation (1), takes into account two of the characteristics of repetitive actions that were identified in section Repetitiveness in Action Recognition, that is (a) the *number of repetitions* or *repetition count*, N_{rep} and (b) a measure of the *variability of repetitions* expressed by the mean repetition similarity μ_{sim}.

For the estimation of the number of repetitions, we exploit the repetition count estimate of *Rep-Net*, by Dwibedi et al. (2020). Regarding the localization of the repetition segments, the boundaries are defined at the frame indices, in which a change in the repetition count occurs, i.e., when *Rep-Net* detects another a repeated instance of the action.

As for the computation of μ_{sim}, this is performed by transforming the TSM, to an affinity matrix, *Sim*. Each cell (i, j) of *Sim* expresses the similarity of encoded repetitions i, j as:

$$Sim(i,j) = e^{-\frac{1}{2}[E(i,j)]^2}, \tag{2}$$

where $E(i, j)$ is the Euclidean distance between the representation of the encoded repetitions i, j. The row-wise mean μ_i, expresses the mean similarity of the ith repetition to the rest. The mean similarity between all repetitions, μ_{sim}, is computed as the mean of μ_i.

Intuitively, the number of repetitions N_{rep} highlights the potential presence of information redundancy due to several instances of the same repetitive segment. The mean repetition similarity μ_{sim} solidifies this by exploring the inter-repetition segment similarities. A high number of repetitions with a high inter-repetition segment similarity indicates that no additional information gains can be obtained by modeling the entire repetition segment set. In this case, we are dealing with repetitive actions that have minimum or no impact on the scene, such as the actions of jumping jacks or clapping, in which the repetitions bear little or no additional information about the action compared to the initial execution. This is manifested with the high similarity among the encoded repetition segments. In such cases, we do not need to pay attention to the features produced by the second sub-net and instead we should shift our interest to the spatio-temporal features that are generated from the RGB sequence of the initial repetition.

On the contrary, a low inter-repetition segment similarity, indicates the potential presence of action-invoked effects on the surround scene and objects such as the actions of wood chopping or onion slicing, for which the effect of the repetitive task (e.g., on the wood plank or on the onion) can be considered as a highly discriminative element. In such cases, our model should consider the features produced by the second sub-net, that is responsible for modeling the inter-repetition segment (long-term) differences.

5. EXPERIMENTS

The performed experiments aim to evaluate, (a) the effect of utilizing repetitions as a means to augment the data bank of an HAR model and constrain the input sequence, (b) the contribution of repetitiveness-based sequence splitting in datasets of repetitive actions with a variety of characteristics, and finally, and (c) the accuracy improvement due to the exploitation of the information regarding gradual scene changes due to repetitiveness. To account for hardware limitations, we sample each sequence using two widely employed window sampling approaches, (a) *window-based uniform sampling* (WS) and (b) *random clip crop sampling* (RCC). In WS, as a key-frame, we select the center frame. For both input sampling schemes, we only utilize the RGB frames, without any embedding generation stage and consider sequence lengths of $\{10, 25, 35, 64\}$ frames for the generated input sequence.

5.1. Datasets

The contribution of the proposed methodology is expected to be more evident in datasets with actions involving repetitive tasks. However, the amount of repetitive actions in the existing datasets varies depending on the complexity and action topic. We examine datasets that consist of (a) repetitive activities, only, with relatively high repetition number and (b) a small percentage of repetitive actions with low repetition number. Both manifest in unconstrained conditions.

Countix (Dwibedi et al., 2020): This is the largest repetitive actions dataset, consisting of in-the-wild videos with challenging recording conditions, such as camera motion, diverse periodicity and repetition ranges (*min* 2 and *max* 73 repetitions), and an average of approximately 7 repetitions per video. In our work, we generated two subsets of *Countix*, with the goal of evaluating (a) the contribution of repetition segmentation as a pre-processing module in HAR and (b) the effect of repetition count and the repetitive action characteristics (presence/absence of gradual effects), on the performance of the proposed HAR model, *RepDI-Net*.

- *CountixHAR*, was generated with a strict repetition count margin (actions with minimum 2 and maximum 10 repetitions) under the constraint that each action class included in the dataset should contain at least 5 samples in order to ensure that a sufficient number of training data will be available for each action class. The resulting *CountixHAR* set comprises 28 action classes and consists of 718 training and 262 test videos[1]. The performed data-pool augmentation (i.e., the consideration of each repetition as a discrete sample) increases the training sample number to 3, 284, indicating that we get on average 4.6 repetitions per sample. This dataset is used to evaluate the contribution of repetition-centered input sequence segmentation in HAR. **Figure 4B**, presents the sequence length distribution change of the samples (train&test) due to the repetition-based splitting.
- *CountixEffects*, was generated to evaluate (a) the impact of the number of repetitions and (b) the contribution of repetition-based segmentation in repetitive actions that impose gradual effects in the environment. *CountixEffects* expands the repetition count range to actions that exhibit up to 20 repetitions, and consists of 5 action classes. Two of them (*sawing wood, slicing onion*) produce gradual effects. The rest 3 actions (*headbanging, doing aerobics, running on treadmill*) do not produce gradual effects. The specific action classes subset of *Countix* were carefully selected so that (a) each action

[1] We only used publicly available YouTube videos.

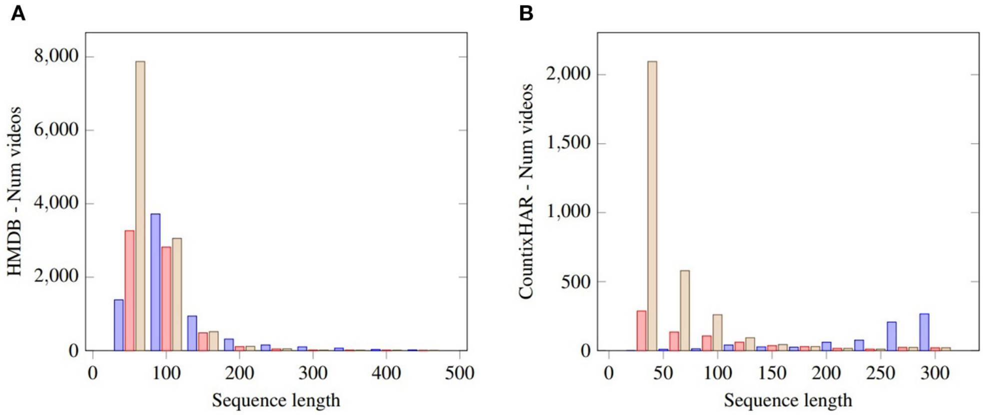

FIGURE 4 | Sequence length distributions according to sequence splitting approaches for **(A)** *HMDB-51* and **(B)** *CountixHAR*. Purple: Initial (whole) sequence, Pink: First execution only, Yellow: All repetitions as distinct samples.

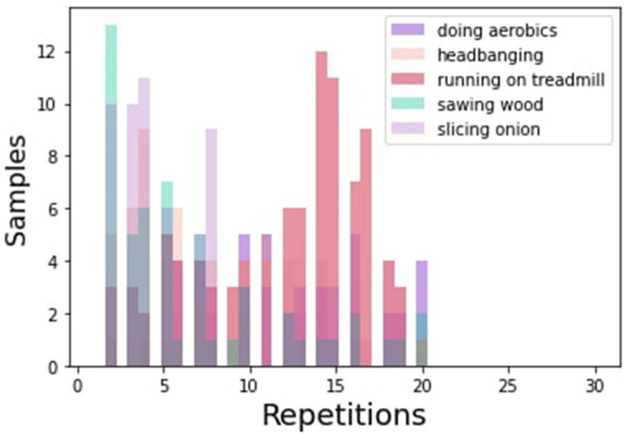

FIGURE 5 | Distribution of video samples in *CountixEffects*, with respect to repetition count.

class contains samples for the majority of repetition counts (above 60%) and (b) there is at least one sample per repetition count. These conditions allowed the generation of a dataset that is fairly balanced with respect to the repetition count. It is noted that the original *Countix* dataset does not possess this characteristic, as it exhibits a right skewed sample per repetition class distribution [**Figure 6** in Dwibedi et al. (2020)]. The resulting *CountixEffects* dataset consists of 322 training and 100 test videos. Based on the ground-truth repetition counts provided in the original *Countix* dataset, the generated subset of *CountixEffects* exhibits an average of 9.67 repetitions per sample (regarding the training subset) and a augmented set of 3,124 training samples. An overview of the training sample distribution per repetition count class for *CountixEffects* is presented in **Figure 5**.

HMDB (Kuehne et al., 2011): This dataset contains 51 action categories, consisting of both repetitive and non-repetitive tasks. The action clips were sourced from movies, YouTube and a number of additional public video repositories. The dataset issues 3 official splits, however, in this work, we only report the top-1 classification accuracy on split 1. This dataset serves as a case of repetition count ground-truth agnostic case, in order to evaluate RepNet's generalization. The initial set consists of 4,697 training, 2,054 test videos and 51 action classes. The augmentation of the data-pool by considering each repetition as a discrete sample increases the training sample number to 8,053, indicating that the initial dataset consisted of action samples with on average 2 repetitions per sample. **Figure 4A**, presents the sequence length distribution change of the samples (train&test) due to the repetition-based splitting.

5.2. Spatio-Temporal DNN Specifications

We utilize the original I3D (Carreira and Zisserman, 2017) design using the pre-trained weights on ImageNet (Deng et al., 2009) and Kinetics (Carreira and Zisserman, 2017), until the last receptive field up-sampling layer-block. As a top-level we include a Convolutional 3D layer (Conv3D) followed by an FC layer with ReLU activation function, plus Batch Normalization, and finally a soft-max activation layer, in order to fine-tune it on the new classification task for the new datasets.

Given the above, the spatio-temporal sequence modeling module of *RepDI-Net* is a two-stream, two branch NN architecture. Both branches follow almost the same design specifications as the baseline model with the following differences: (a) the number of channels of the Conv3D layer for the sub-net that uses the encoded repetition sequences are 1/4 of the ones used in the reference execution sub-network, (b) the output tensors of their FC layers (ReLU) are concatenated and the resulting tensor is passed into a set of two FC layers (ReLU and soft-max), and (c) the output tensor of the FC layer (ReLU) of the encoded repetitions sub-net is passed through a feature scaling layer (multiplication layer) that utilizes the scaling factors, mentioned in section REPDI-NET: A Deep Architecture to Exploit Action Repetitiveness.

5.3. Training Configurations

For repetition counting/segmentation we exploited the *RepNet* model (Dwibedi et al., 2020), without any dataset-specific fine-tuning, using the off-shelf weights. It should be noted that the documentation of *Countix* does not provide the starting and ending frame indices of each repetition segment. The only available information is the repetition counts. According to Dwibedi et al. (2020), the performance of RepNet in estimating the correct repetition count, under the Off-by-One (OBO) repetition count error metric, leads a 0.3034 miss-classification error for the *Countix* test set. For the *CountixHAR* subset the miss-classification error has been found to be 0.4030 for the combined train and test splits. For *HMDB*, RepNet was applied in a repetition-agnostic manner.

For *HMDB* we applied the standard training/validation/test splits followed in the HAR literature. For *CountixHAR*, and, *CountixEffects*, we defined the dataset training/test split relying on the training/validation/test splits provided in the original *Countix* dataset, with the difference that the validation set was used in the place of the test set, since the test set of *Countix* does not provide any action labels. This resulted in a train-test split without the presence of a validation set.

The action recognition DNNs use the Adadelta optimizer, a learning rate of 0.01, with learning rate decay of $1e-4$, and batch size 8 for $10, 25, 35$ sequence lengths, and batch size 4 for a sequence length of 64. Input sequence length for the encoded repetition sub-net is set to 10 frames (max repetition number). For samples with fewer repetitions we duplicate (not loop) the DIs to the desired length. We did not utilize standard data augmentation schemes, such as horizontal flipping, zooming or region cropping. During testing, with the exception of *RepDI-Net*, we use the original test sets, without repetition localization and segmentation. For *RepDI-Net*, test samples were segmented based on repetition, and then used for the computation of the repetition segments DIs.

6. EXPERIMENTAL RESULTS

We present an evaluation of the impact of repetition segmentation in HAR in relation to the characteristics of the repetitive actions of the employed datasets. We proceed with a series of experiments that demonstrate the importance of correct repetition localization, when exploiting action repetitiveness. Finally, a series of experiments are presented that examine the contribution of key components in the proposed repetition-centered HAR deep architecture, to highlight the benefits and constraints of the proposed pipeline.

6.1. Effect of Repetition Segmentation on HAR Accuracy

We present experiments performed on *CountixHAR* and *HMDB* to evaluate the utilization of a repetition-centered segmentation module as a pre-processing step of input sequence configuration, and its impact on HAR. In **Table 1**, we observe that considering only the initial action execution (1st, 5th rows) reduces the processing cost with an accuracy loss around 1% for sparsely and $2-3\%$ for densely sampled sequences as opposed to using the entire sequence (2nd, 6th rows). This result indicates that for repetitive tasks, each action repetition contains similar information regarding the general action pattern. The score difference between the cases where the entire sequence or only the first execution is considered can be potentially attributed to the long-term action effects on the scene or the object of interest. In addition, the utilization of all repetition segments as discrete samples (3rd, 7th rows) allows for an increase between $2-4\%$ in recognition accuracy, but with an additional computational cost during learning. This strategy is more beneficial, for datasets with highly repetitive actions (i.e., *CountixHAR*), with stronger contribution when using a sliding window sampling, as opposed to the utilization of a random clip cropping strategy for input sequence configuration. This is attributed to the fact that in this strategy, when sampling the entire video, the sparseness of the sampling process is likely to disrupt the action step temporal ordering due to the repetitive nature of the action. The impact of this is more severe in dataset cases with fewer repetitions per action (i.e., *HMDB*), due to the additional temporal ordering disruption cases produced by potentially erroneous repetition segmentations. The effect of the latter factor in the overall HAR accuracy is further assessed later in this section.

Moreover, as shown in **Table 1**, the utilization of dual-stream HAR DNN, in which the first branch is introduced with the most representative execution segment of the repetitive action sample (RGB sequence), and the second with a sequence of encoded frames, each corresponding to a summarization of a single repetition, allows to more effectively represent the discriminative information of the repetitive action. Specifically, in **Table 1** (4th, 8th rows), we observe that this strategy improves accuracy by $1-3\%$ compared to usage of the entire sequence (2nd, 6th rows), for small to mid-range inputs. The improvement in recognition accuracy is observed for both sampling schemes that are used in this study (uniform window sampling—WS, random clip cropping—RCC) for datasets that exhibit moderate to high number of repetitions, such as *CountixHAR*, and for sparse and moderate sampling densities of the input sequence. In datasets with low numbers of repetition such as *HMDB*, the proposed approach improves recognition accuracy, compared to the utilization of a naive sampling strategy on the entire sequence, only for the case of a uniform window-based sampling schemes, with the improvement being observed for sparsely to moderate sampling densities. When a random clip cropping sampling scheme is followed, the proposed approach exhibits lower performance. This is attributed to errors in the repetition segmentation process.

In the case of highly repetitive actions, *RepDI-Net* is capable of decoupling the main action pattern and the action-invoked effect on the scene/objects during learning. This is illustrated in **Figure 6** where we used Grad-CAM (Zhou et al., 2016; Selvaraju et al., 2017) to visualize the activation maps of *RepDI-Net* for each input source as well as the ones of the baseline model that uses a sampled version of the entire video sequence. In the two Countix-sampled cases, the *RepDI-Net*'s RGB input corresponding to a reference action execution focuses on the

TABLE 1 | Accuracy (%) for HAR in HMDB / CountixHAR for different methodological variants.

	HMDB / CountixHAR	*10 frms*	*25 frms*	*35 frms*	*64 frms*
1	*Rep_0-WS*	51.38 / 50.35	57.63 / 59.22	61.82 / 58.87	63.39 / 62.95
2	*All frames-WS*	52.18 / 51.79	58.39 / 59.60	57.59 / 61.51	**66.09** / 63.36
3	*Rep_{all}-WS*	52.08 / 53.25	59.21 / **63.32**	**63.03** / **64.41**	64.20 / **67.04**
4	R_{euc}, DIs, R_{sc}-WS	**53.45** / **54.92**	**59.24**/ 60.22	62.17 / 62.87	63.46 / 63.47
5	*Rep_0-RCC*	49.14 / 53.40	58.74 / 59.44	60.17 / 57.39	63.05 / 58.75
6	*All frames-RCC*	**52.08** / 55.29	**60.36** / 60.22	**62.24** / 62.12	**65.32** / 64.39
7	*Rep_{all}-RCC*	51.80 / **57.95**	59.28 / **64.15**	61.91 / **64.59**	63.62 / **65.48**
8	R_{euc}, DIs, R_{sc}-RCC	49.61/ 56.18	58.78 / 62.35	59.66 / 63.87	61.83 / 63.89

WS, window sampling; RCC, random clip crop. Columns refer to sampled input sequence length for the reference execution and the initial sequence. Bold values indicates the best performing method.

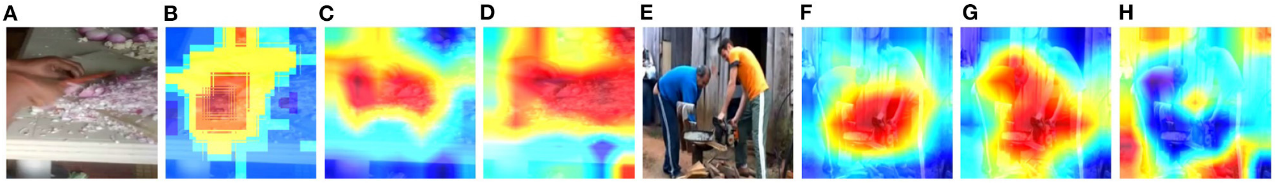

FIGURE 6 | Visualization of class activations for 1) *Onion slicing* **(A–D)**, 2) *Wood sawing* **(E–H)**. In each case, *1st image*: original frame, *2nd image*: using a window sampled sequence of the original video, and *3rd, 4th images*: reference RGB execution and DIs sources, respectively.

main motion pattern of the action, whereas the repetition-oriented part focuses on regions around the main motion pattern region. We would expect the HAR model to focus on regions that the action effects explicitly (around the object of interest) and not in action-unrelated regions. This is indeed true for actions such as onion slicing (**Figure 6D**). Our model is not able to exhibit similar behavior for sequences with sudden viewpoint changes and severe occlusions such as the wood sawing sequence in *CountixHAR* (**Figure 6H**). To explain this behavior we should consider that the original *Countix* dataset contains real-world videos with samples that exhibit sudden viewpoint changes and severe occlusions. In such cases, the examined repetition summarization technique (DIs) reduces but does not eliminate these effects. This "noise" in the data leads the DI sub-net to expand the range of the regions that focuses on.

6.2. Impact of Repetition Segmentation Performance on Recognition Accuracy

The performance of an HAR model that exploits the repetitive nature of certain actions is expected to depend on the accuracy of the repetition segmentation task. Incorrect segmentation of the input actions can potentially disrupt the action step ordering that is encapsulated within each segmented repetition. As stated earlier, the *Countix* dataset provides only the estimated repetition counts of the included action samples, and does not include any information on the repetition segment start/end frame indices. Under those conditions, we design our experiments by focusing on the correctness of the repetition count estimate, considering that in each case there exist discrepancies between the estimated and expected repetition segment boundaries. Consequently, the cases that are considered are the following (see **Figure 7** for a graphical illustration):

Correct repetition count, unverified segment duration: To simulate the effect of unverified and perhaps erroneous repetition segmentation and assess its impact on the proposed DNN architecture for the case of *CountixHAR*, we uniformly segment the sample sequences into the estimated number of repetition segments that are provided in the dataset documentation. By comparing the performance of *RepDI-Net*, using this segmentation strategy against its performance with the utilization of *RepNet* as the repetition segmentation approach, we observe that erroneous localizations have a negative effect on the recognition accuracy, resulting in a decrease in accuracy, between $1.5 - 3\%$, as shown in **Table 2** (row 1 vs. row 2, row 4 vs. row 5). This accuracy drop is more pronounced for sparse sampled sequences, in which redundant or absent action data has a larger impact on the classification, since discriminative keyframes can be discarded.

Incorrect repetition count, unverified segment duration: In this case, we randomly assign a value for the repetition count number and sample within the range of estimated repetition counts of each class. Moreover, to further increase the possibility of erroneous repetition segment duration estimates, we split the sequence into the assigned number of repetitions with a variable repetition segment size. As previously, we compare the performance of *RepDI-Net*, with this splitting strategy. Our results, shown in **Table 2** (row 1 vs. row 3, row 4 vs. row 6), indicate a decrease in performance between 6 to 15%. Higher performance drops are observed for sparser sampled

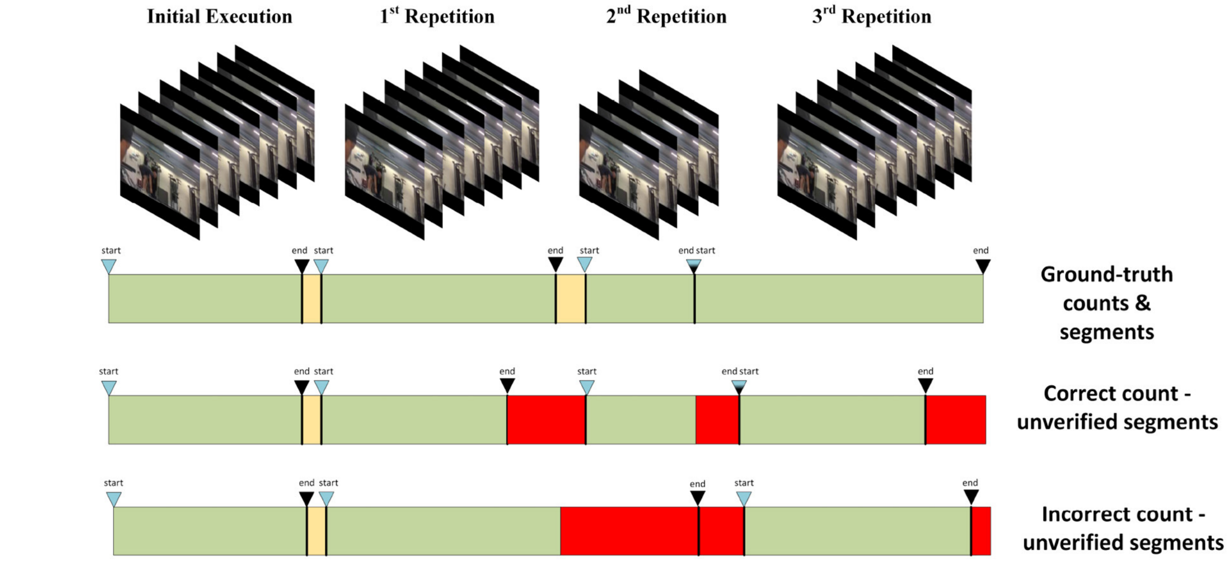

FIGURE 7 | Possible outcomes in repetition detection. A *yellow box* denotes a background action class segment (i.e., an action that is not part of the repetitive action), a *red box* denotes an erroneously segmented part, and a *green box* denotes a correct detection. First line: the ground truth on the number of the segments and their start/end frames. Second line: the number of repetitions is correctly estimated but the repetition segment boundaries are not. Third line: Both the number and the boundaries of repetition segments are not correct.

TABLE 2 | Accuracy (%) for CountixHAR (1, 4) using *RepDI*, (2, 5) correct repetition count but unverified repetition segment duration, and (3, 6) incorrect repetition counts and unverified repetition segment duration.

	Repetition segmentation scheme	*10 frms*	*25 frms*	*35 frms*	*64 frms*
1	RepNet Dwibedi et al. (2020)-WS	**54.92**	**60.22**	**62.87**	**63.47**
2	Correct rep. count/Unverified segment-WS	52.27	58.84	60.98	61.60
3	Incorrect rep. count/Unverified segment-WS	46.39	46.15	47.23	52.49
4	RepNet Dwibedi et al. (2020)-RCC	**56.18**	**62.35**	**63.87**	**63.89**
5	Correct rep. count/Unverified segment-RCC	53.51	59.44	59.09	62.12
6	Incorrect rep. count/Unverified segment-RCC	44.30	48.92	55.09	57.38

Results are provided for both the WS (1, 2, 3) and RCC (4, 5, 6) schemes. Bold values indicates the best performing method.

input sequences, where an erroneous segmentation increases the possibility of selecting a frame that does not maintain the temporal order consistency of the action steps. Both experimental scenarios highlight the importance of an accurate repetition segmentation methodology.

From the aforementioned results, it can be observed that, we obtain better results if we rely on the (possibly wrong) number of repetitions estimated by RepNet, compared to the application of naive segmentation and sampling approaches. Therefore, it is expected that an improvement in the performance of the repetition count estimation and segmentation module will further improve the accuracy of an HAR model.

6.3. Effect of Repetition Temporal Encoding on Recognition Accuracy

The effectiveness of the repetition-driven sub-net in the proposed pipeline depends on the ability of the temporal encoding method that is used to represent the discriminative elements in each repetition. To better examine the importance of this factor, we compared the contribution of the Dynamic Images (DIs) encoding against a deep learning encoder, by examining the effect on the recognition accuracy. In the place of the deep temporal encoder, I3D was exploited, following similar directions in the HAR literature that exploit 3D Convolution-based encoders (Wang et al., 2016; Lin et al., 2019). Experimental results shown in **Table 3** (row 1 vs. row 3, row 4 vs. row 6) indicate that the repetition-oriented sub-net benefits from a more informative representation, with improvements in the range of $1-4\%$.

6.4. Effect of Scaling Factor on Recognition Accuracy

Table 3 (row 1 vs. row 2 and row 4 vs. row 5) presents the effect of the scaling factor, on the recognition accuracy. We observe that if this scaling factor is not employed, accuracy decreases significantly (between 2.3 and 4.9%). The proposed scaling factor

TABLE 3 | Accuracy (%) for CountixHAR, due to (A) scaling factor absence (*No R_{sc}*), and (B) substitution of Dynamics Images (DIs) with a deep-based (I3D) repetition temporal encoder (*I3D Enc*).

	CountixHAR	*10 frms*	*25 frms*	*35 frms*	*64 frms*
1	R_{euc}, DIs, R_{SC}-WS	54.92	60.22	62.87	63.47
2	R_{euc}, DIs, NoR_{SC}-WS	50.02	57.91	59.22	60.89
3	$R_{euc}, I3DEnc, R_{SC}$-WS	**55.27**	**62.12**	**63.66**	**64.36**
4	R_{euc}, DIs, R_{SC}-RCC	56.18	62.35	63.87	63.89
5	R_{euc}, DIs, NoR_{SC}-RCC	53.72	58.09	61.57	61.66
6	$R_{euc}, I3DEnc, R_{SC}$-RCC	**61.60**	**63.33**	**64.39**	**65.52**

Bold values indicates the best performing method.

TABLE 4 | Accuracy on *CountixEffects*, due to the presence of the scaling factor f_{DIsc}, for actions (A) without any effects on an object due to repetitiveness, NoR_{sc}, (B) with effects on an object due to repetitiveness, R_{sc}.

CountixEffects action subset	NoR_{sc}	R_{sc}
No gradual effect set 2-7	81.00	86.33
Gradual effect 2-7	73.00	79.50
No gradual effect set 8-20	80.33	88.67
Gradual effect set 8-20	76.00	78.00

Table also depicts the effect of repetition count (2–7, 8–20) on each subset.

TABLE 5 | Accuracy on *CountixEffects*, for lower (2–7) and higher (8–20) repetition counts, using a 10-frame input sequence, and a uniform sampling scheme, (A) using all frames (B) using the *RepDI* approach and (C) using all repetitions as samples.

CountixEffects action subset	(A) *All frames*	(B) *RepDI*	(C) *Rep_{all}*
Repetitions count 2-7	75.67	82.91	85.40
Repetitions count 8-20	80.80	83.34	88.60

Approaches (B), (C) exploit repetitiveness and outperform (A) that does not.

TABLE 6 | Time per epoch (sec—mean duration over 5 epochs).

Dataset, Learning method	Time (sec) per epoch
CountixHAR 2-10, *Rep_{all}*	548
CountixHAR 2-10, RepDI	125
CountixEffects, 2-20 *Rep_{all}*	521
CountixEffects, 2-20 RepDI	62

Training was performed in an RTX 3070 GPU, batch size 8, learning rate 0.8, Adadelta, input length 10 frames, RCC sampling.

tunes the representation learnt by the model according to the variability of the repetitive segments and the potential presence of discriminative information among them. This feature can be of importance when learning to recognize actions that impose gradual effects on the scene. To better evaluate the importance of f_{DIsc} with respect to the repetition count range, we formulate our experimental set up around *CountixEffects* as follows. We splitted *CountixEffects* into two subsets based on different repetition ranges, (a) *limited repetitiveness*, consisting of samples whose repetition count was in the range of $[2, 7]$ and (b) *moderate to high repetitiveness*, whose repetition count was in the range of $[8, 20]$. This formulation resulted in the subset $2 - 7$, to consist of 134 training samples, whereas the subset $8 - 20$, to contain 189 training videos. For both subsets a common test set was created consisting of 100 videos with repetition counts in the range of 2–20. Specifically, 64 test samples lie within the 2–7 repetition range and 36 samples within the 8–20 range. Moreover, we examine the performance of *RepDI-Net*, for repetitive actions (a) with *no notable effect on the scene* and (b) *a gradual effect on the scene*, by leveraging the action set layout of *CountixEffects*. Our experimental results, shown in **Table 4**, indicate that the inclusion of the repetition count and similarity-driven scaling factor for the repetition-based DI branch is beneficial for actions that impose a gradual effect on the scene. Moreover, for both repetition count ranges, the scaling factor improves recognition accuracy in both action subsets.

Table 5 shows the accuracy on *CountixEffects* using a 10-frame input sequence, and a uniform sampling scheme, (a) using all frames, (b) using the *RepDI* approach, and (c) using all repetitions as samples. Results are reported for the cases of actions in two different repetition ranges (2–7 and 8–20). As it can be verified, the application of a repetition-based segmentation stage [columns (b) and (c)] leads to an improvement in recognition accuracy compared to a naive use of the input sequence [column (a)]. Moreover, the consideration of each repetition as a distinct sample performs better compared to the *RepDI* approach. However, as presented in the following section, *RepDI* involves a more compact representation of the input sequences and, as such, is computationally more efficient compared to using all repetitions. This is because when all repetitions are used, the training time is proportional to their number.

6.5. Learning Efficiency When Exploiting Action Repetitiveness

In the previous experiments it is clear that the exploitation of action repetitiveness for better configuring the input sequence can improve the model performance in HAR. Based on the obtained experimental results, the most effective approach is to consider all repetitions as distinct training samples. However, this approach is not the most efficient since the training time per epoch increases proportionally to the repetition count. As shown in **Table 6** for the case of *CountixHAR*, a mean of 4.7 repetitions per sample leads to a ×4 increase in the per epoch computation time compared to exploiting the repetitions with the proposed deep pipeline, with the computation time discrepancy increasing proportionally as the number of repetitions increases. Moreover,

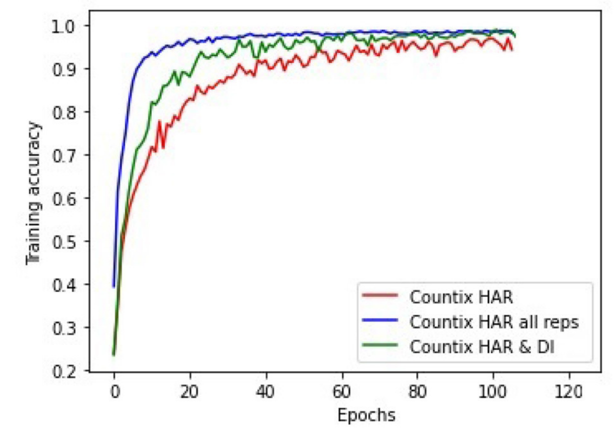

FIGURE 8 | Training accuracy change with (A) no exploitation of the repetitiveness (red), (B) exploitation of all repetitions as distinct samples (blue), and (C) using the proposed method (green).

based on **Figure 8** it is evident that the proposed repetition segment summarization scheme, achieves the best trade-off regarding efficiency and efficacy in the learning process when dealing with repetitive actions.

7. DISCUSSION AND FUTURE WORK

We considered and evaluated the repetitive nature of certain actions in HAR under two perspectives. First, we investigated the effect of redundant information presence, due to task repetitiveness, on the ability to learn discriminative action-specific representations using common sampling techniques. Additionally, we proposed ways to highlight, *via* effective repetition sequence localization and processing, the gradual effects of the repetitive action on the actor or on the involved objects, and evaluated their contribution/importance on the action recognition task. Our findings indicate that for actions exhibiting moderate to high number of repetitions, localizing and using repetitions allows a deep learning HAR model to access more informative and discriminative representations, thus, improving the recognition performance. Exploiting repetitions as discrete samples leads to slower learning rates but allows the model to better capture the temporal ordering of the action as well as the scene/actor/object-of-interest appearance changes.

Repetitions can also be used to highlight the gradual effects of the action on the scene, an ability that can be useful to discriminate between *fine-grained* actions that exhibit high appearance and motion similarities. When adopting this strategy, it is evident that HAR should focus on the action-affected regions. Our findings indicate that the presence of action background motions or occlusions that are unrelated to the action, tend to be captured by summarization methods and are, therefore, being considered as action-induced consequences. A remedy to this could be to focus on the action-related objects and generate temporal summarizations/encodings only for these regions. This encoding scheme, when accompanied with the use of more recent state-of-the-art deep HAR models, will allow for more informative representations, and is expected to increase the effectiveness of exploiting repetitiveness on the HAR task.

One of the most important issues in the direction of exploiting repetitiveness in HAR, is the accuracy in repetition localization. This is still open for improvement, since only a few works have tackled the problem, all of them under the perspective of periodicity estimation and repetition counting. As indicated by our experiments, an HAR model that exploits repetitiveness is expected to benefit from a more robust repetition localization and repetition count estimation method.

DATA AVAILABILITY STATEMENT

In this study, the authors generated and analyzed two subsets of Countix (Dwibedi et al., 2020), dubbed CountixHAR, and CountixEffects. Guidelines to generate the dataset, as well as any important documentation, can be found in the github repository Repetitive-Action-Recognition: https://github.com/Bouclas/Repetitive-Action-Recognition.

AUTHOR CONTRIBUTIONS

KB introduced the conceptualization, implemented the software, performed the experiments, and wrote most of the paper. AA assisted in refining the methodology, validating the results, and contributed to the writing of the paper. All authors have read and agreed to the published version of the manuscript.

FUNDING

This research project was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I) under the 1st Call for H.F.R.I Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment Project I.C.Humans, Number: 91.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of NVIDIA Corporation with the donation of a GPU. The authors would like to thank Kostas Papoutsakis and Aggeliki Tsoli for their helpful suggestions.

REFERENCES

Abdi, H., and Williams, L. J. (2010). Principal component analysis. *Wiley Interdiscipl. Rev. Comput. Stat.* 2, 433–459. doi: 10.1002/wics.101

Aggarwal, J. K., and Ryoo, M. S. (2011). Human activity analysis: a review. *ACM Comput. Surveys (CSUR)* 43, 1–43. doi: 10.1145/1922649.1922653

Ahad, M. A. R., Tan, J. K., Kim, H., and Ishikawa, S. (2012). Motion history image: its variants and applications. *Mach. Vis. Appl.* 23, 255–281. doi: 10.1007/s00138-010-0298-4

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*.

Bilen, H., Fernando, B., Gavves, E., and Vedaldi, A. (2017). Action recognition with dynamic image networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 2799–2813. doi: 10.1109/TPAMI.2017.2769085

Carreira, J., and Zisserman, A. (2017). "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI), 4724–4733.

Cherian, A., Fernando, B., Harandi, M., and Gould, S. (2017). "Generalized rank pooling for activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 3222–3231.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL: IEEE), 248–255.

Diba, A., Sharma, V., and Van Gool, L. (2017). "Deep temporal linear encoding networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 2329–2338.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., et al. (2015). "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 2625–2634.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. (2020). "Counting out time: class agnostic video repetition counting in the wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA).

Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 1933–1941.

Fernando, B., Gavves, E., Oramas, J. M., Ghodrati, A., and Tuytelaars, T. (2015). "Modeling video evolution for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 5378–5387.

Herath, S., Harandi, M., and Porikli, F. (2017). Going deeper into action recognition: a survey. *Image Vis. Comput.* 60, 4–21. doi: 10.1016/j.imavis.2017.01.010

Kang, S. M., and Wildes, R. P. (2016). Review of action recognition and detection methods. *arXiv*. Available online at: https://arxiv.org/abs/1610.06906

Karvounas, G., Oikonomidis, I., and Argyros, A. (2019). Reactnet: Temporal localization of repetitive activities in real-world videos. *arXiv preprint* arXiv:1910.06096.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). "Hmdb: a large video database for human motion recognition," in *2011 International Conference on Computer Vision* (Barcelona: IEEE), 2556–2563.

Levy, O., and Wolf, L. (2015). "Live repetition counting," in *2015 IEEE International Conference on Computer Vision (ICCV)* (Santiago), 3020–3028.

Lin, J., Gan, C., and Han, S. (2019). "Tsm: temporal shift module for efficient video understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 7083–7093.

Lin, R., Xiao, J., and Fan, J. (2018). "Nextvlad: an efficient neural network to aggregate frame-level features for large-scale video classification," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. Available online at: https://openaccess.thecvf.com/content_eccv_2018_workshops/w22/html/Lin_NeXtVLAD_An_Efficient_Neural_Network_to_Aggregate_Frame-level_Features_for_ECCVW_2018_paper.html

Panagiotakis, C., Karvounas, G., and Argyros, A. (2018). "Unsupervised detection of periodic segments in videos," in *2018 25th IEEE International Conference on Image Processing (ICIP)* (Athens), 923–927.

Pearson (1901). On lines and planes of closest fit to systems of points in space. *London Edinburgh Dublin Philosoph. Mag. J. Sci.* 2, 559–572.

Runia, T. F. H., Snoek, C. G. M., and Smeulders, A. W. M. (2018). "Real-world repetition estimation by div, grad and curl," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Available online at: https://openaccess.thecvf.com/content_cvpr_2018/html/Runia_Real-World_Repetition_Estimation_CVPR_2018_paper.html

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). "Grad-cam: visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 618–626.

Simonyan, K., and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *arXiv preprint* arXiv:1406.2199.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision* (Santiago), 4489–4497.

Varol, G., Laptev, I., and Schmid, C. (2017). Long-term temporal convolutions for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 1510–1517. doi: 10.1109/TPAMI.2017.2712608

Wang, L., Tong, Z., Ji, B., and Wu, G. (2021). "Tdn: temporal difference networks for efficient action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN), 1895–1904.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L. (2016). "Temporal segment networks: towards good practices for deep action recognition," in *European Conference on Computer Vision* (Amsterdam: Springer), 20–36.

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 7794–7803.

Zhang, S., Guo, S., Huang, W., Scott, M. R., and Wang, L. (2020). V4d: 4d convolutional neural networks for video-level representation learning. *arXiv preprint* arXiv:2002.07442.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV,) 2921–2929.

Zhu, Y., Li, X., Liu, C., Zolfaghari, M., Xiong, Y., Wu, C., et al. (2020). A comprehensive study of deep video action recognition. *arXiv preprint* arXiv:2012.06567.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Bacharidis and Argyros. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.