Repetition-aware Image Sequence Sampling for Recognizing Repetitive Human Actions

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

[1]Computer Science Department, University of Crete
[2]Institute of Computer Science, FORTH,
{kbach, argyros}@ics.forth.gr

Abstract

In the field of video-based human action recognition (HAR), standard hand-crafted and deep learning-based approaches are constrained by the computational and memory requirements of their models and the length of the input sequence that can be processed during learning. Sampling techniques employing a windowed or a random clip cropping have been the simplest and most effective ways to cope with limitations on the maximum possible length of the input sequence. However, such designs do not guarantee that the correct ordering of the action steps is captured, or require several learning iterations. In this work we address this problem for the class of repetitive actions. Specifically, given a temporal segmentation of a repetitive action into its repetitive segments, we propose and develop novel approaches for ranking and selecting/sampling segments so as to improve learning in deep models for HAR. We show that by employing the proposed repetition-aware sampling schemes in state-of-the-art deep models for HAR, the action recognition accuracy is increased. The proposed approach is evaluated on existing datasets and on a new dataset that is tailored to the quantitative evaluation of the task at hand. The obtained results reveal how our approach performs in relation to various characteristics of the observed repetitive actions (repetition frequency, their effects on scene objects, etc) and demonstrate the performance improvements.

1. Introduction

Human action recognition (HAR) is a crucial perceptual component for the development of robotic systems that are able to interpret and understand human actions and predict future ones. HAR enables robots to respond to human actions enhancing task performance and the possibility of human-robot interaction in real-world scenarios. Thus, HAR facilitates the seamless integration of robots into human-centric environments.

Human actions exhibit high variability in their durations, execution style, temporal orderings which results in diverse appearance and motion characteristics [1, 11]. The extent of these variations also depends on the nature of the action and the scene context in which it takes place. To deal with these complexities, HAR models attempt to assimilate both short and long-term action-related information. At the same time, coping with arbitrarily long temporal information has a negative impact on the computational footprint of the model. In order to deal with this trade-off, the length of the input frame sequence is constrained using temporal sequence sampling and clip (segment) encoding techniques. Existing sampling strategies in HAR methods (deep learning, hand-crafted and hybrid [11, 36]) tend to not always maintain the temporal ordering of the action steps. This is mainly due to the temporal duration variations of the actions and the presence of video segments that record actions that are irrelevant to the one that needs to be recognized. This has a negative impact in the effectiveness of HAR, especially when distinguishing between actions that exhibit similar appearance, motion and execution characteristics.

While a general solution to the problem is still lacking, we foresee that there is a lot of space for improvement in the very interesting and quite common case of *repetitive actions*. A number of techniques [23, 7] are capable of deciding whether an action is repetitive or not *without recognizing its action class*. Moreover, they can count the repetitions and segment them temporally in the input sequence. We claim that a repetition-depended input sequence segmentation would remove redundant information during the sampling process (i.e., due to repetition segment similarities) or irrelevant information (i.e., due to other, irrelevant actions appearing before, after or between the repetition segments).

Given the above-mentioned observation, in this work,

*Corresponding author

we develop such repetition-aware input sequence sampling strategies and investigate their potential impact on the input configuration of HAR deep models. In addition, we show how the temporal localization of the repetitive segments allows the spatio-temporal detection of a possible effect that the action has on the appearance and state of the actor and/or the object(s) that participate in it. Finally, we propose a set of intuitive modifications of the structure of HAR deep models that enable them to exploit repetition-relevant information sources to significantly improve their accuracy. The analysis of the obtained quantitative results in a series of experiments in several well-established datasets and on a new dataset that we introduce in this work, demonstrates the significant accuracy gains achieved by the proposed approach.

2. Related Work

Action recognition & sequence sampling: The temporal capacity of HAR models, even in the deep learning era, does not scale with the range of the temporal complexities, ordering, appearance and duration variations of video recordings of human actions. To address this limitation, HAR methods apply input video/sequence sampling to constrain the computational load and memory requirements by processing only salient video parts. The majority of recent deep learning HAR methods still rely on simple sparse or dense sampling routines. Sampling is performed at a frame level over the entire image sequence or at a segment (clip) level. The sampled frames (or segments) are forwarded to the model either in the raw RGB format (entire sequence [25, 5, 8], segment-wise [28, 27]) or in a deep encoded representation generated from pretrained image and video deep models (2D-CNN models at a frame-wise level [33, 26], or 3D-CNN models at a segment-wise level [34, 29]). Each sampling density favors the short- or long-range modeling of the action dynamics, and is always related to the temporal modeling capacity of a certain model.

Despite their computational efficiency, these sampling schemes do not guarantee that the discriminative stages/steps of an action and/or their temporal order is maintained. Moreover, treating every video frame/clip equally for the model's input configuration also allows the consideration of frames/clips that are irrelevant to the labeled action category, thus negatively affecting the learning process. To alleviate this problem, a set of methods proposed to sample the frames/clips containing the most informative or discriminative parts of the sequence. The works of Wu *et al*. [30] and Korbar *et al*. [18] structure their proposed sampling schemes based on frame information importance ranking operations of modern video compression representations, and train their HAR deep models on the compressed video representations. Contrary, another class of approaches incorporates the task of key-frame/clip identification into the action recognition problem by also scoring the importance of frames/clips on the classification outcome, with supervised [14] or reinforcement learning schemes [31].

Detection & counting of repetitive actions: The proposed approach for repetition-aware HAR capitalizes on the availability of methods that are able to segment the repetitive segments of repetitive actions. A number of methods have dealt with the topic of periodicity/repetitiveness detection and classification in video data, evaluating their success by measuring their accuracy in repetition counting. Towards this end, the most common strategy followed in the literature for estimating the number of repetitions is to detect the set of repetitive segments in the frame sequence by examining frame-wise/clip-wise correlations. This is achieved by constructing a Temporal Self-similarity Matrix (TSM). In TSMs, each frame/clip is encoded using hand-crafted [22, 15] or deep [7, 12] features. The repetitiveness estimates are then generated by casting the problem as a shortest path estimation problem with graph-based methods applied to the TSM [22], or as a multi-class classification task in deep models [19, 7, 15, 12], with each class corresponding to a different repetitive segment length. Finally, another set of recent works exploit Fourier or Wavelet analysis [23] to detect repetitiveness in 1D signals generated from frame-wise motion or appearance feature descriptors.

Our contribution: The present work serves as an extension of our earlier research [3], which introduced a foundational dual-branch neural network architecture utilizing repetition-focused input segmentation for HAR. In this continuation, our objective is to refine and advance this framework, while also demonstrating its adaptability to various established deep learning HAR models. To elaborate, the main contributions of this study can be outlined as follows:

- We propose a novel approach for exploiting repetition-aware information in HAR, by contributing a new ranking scheme based on segment-wise differences that highlights the importance of each repetition segment on the recognition of the repetitive action. This is combined with a novel subnet design that, compared to [3], captures more effectively the appearance changes on scene elements, as they are inflicted by the execution of the repetitive action.

- We demonstrate that our new proposal for repetition-aware segment sampling, when incorporated into two state-of-the-art HAR deep models (TDN [27], SlowFast [8]), results in significant accuracy improvements.

- We introduce a new HAR dataset that contains annotations of the temporal boundaries of the repetition segments, thus allowing the quantitative evaluation of techniques that exploit repetitiveness for HAR.

3. Action Repetitiveness

In repetitive actions, the execution of each repetitive segment has the same structure, with potential variations being observed in the execution tempo and on the effects of the action on the scene and the involved objects. The presence of task repetitiveness can be considered as an indicator of information redundancy since the motif of the action can be recognized by only viewing a single repetitive segment of the task, as for example in the action of *jumping jacks*. However, some repetitive actions may have an effect on the actor and/or the surrounding scene as, for example, in the case of *slicing a fruit/vegetable*. As far as action recognition is concerned, these gradual changes in the scene elements may contain information of strong discriminative power. This is particularly true for the case of actions that share similar motion motifs and appearance characteristics, e.g. the actions of *slicing* and *dicing* a vegetable/fruit.

An additional characteristic of repetitive actions is that the number of repetitions is rarely known a priori. Any action can become repetitive if it is repeated within the same temporal segment more than once, and the repetition counts can differ between different samples of the action. Ideally, we would like a HAR method to be able to process all the repetitions of the action in order to learn the discriminative information regarding the action's structure, motion motif and effect on the surrounding space. An important arising question is whether *it is necessary for a model to process every execution of the action in order to learn recognize that action*. The answer is not obvious as it depends on the action complexity and context. For simple, repetitive actions with highly discriminative motion or appearance cues, such as *clapping*, a single or a few repetitions might be enough. On the other hand, for complex and fine-grained actions, such as *slicing*, a more elaborate repetition segment selection is required to capture the action's intrinsic complexity.

4. Proposed Approach

We propose HAR strategies and modifications to existing HAR architectures to effectively exploit action repetitiveness during the definition of their input. The goal is to highlight both the execution steps and gradual effects of the action (if any). Firstly, we present a process that allows for a compact and meaningful representation of the gradual effects of the repetitive execution of the action. Then, we propose a set of augmentations and modifications on two mainstream deep HAR models to better exploit the information content of an input under a repetition-aware perspective.

We assume the availability of a mechanism that can perform a temporal segmentation of a repetitive action into its repetitive segments. As stated in Section 2, there exist a number of unsupervised methods that allow for a temporal segmentation of the input sequence based on task repetitiveness. In our work we rely on RepNet by Dwibedi *et al*. [7] to define each repetition's temporal boundaries.

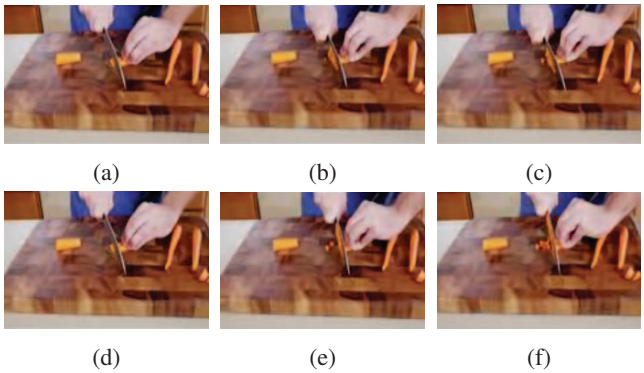


Figure 1: Example of repetition selection under a segment processing capacity of 3 segments. The input video shows a dicing carrot action. The repetition-centered temporal segmentation of the video was performed manually, and produced 13 segments. (a)-(c): the last frames of the first 3 executions following a simple ordered selection. (d)-(f): the last frames of the 1st, 10th, and 13th executions, of the 3 segments selected with the proposed KKZ [16] approach.

4.1. Capturing action effects on the scene

Splitting the input sequence based on task repetitiveness allows the decoupling of the execution steps, motion and coarse appearance characteristics of the action from the gradual action effect on the scene and objects-in-use. The potential effect of the execution of each repetitive segment is manifested with shape deformations and appearance variations on the object/person/scene elements that are involved in the action. Such changes are characteristic of the action, and may be very important for discriminating between *fine-grained* action classes. This means that given an action segment with a single execution, the sampled frames should encapsulate (a) the action steps, and, (b) the action effects. This process is not trivial since the temporal execution tempo in realistic conditions varies in each action sample as well as between each execution repetition for a specific sample. This, in turn, means that a sampling approach that is based on a fixed-size window might be suboptimal.

In order to effectively capture the consequences of an action onto a scene, we can utilize video sequence summarization or temporal encoding and rank pooling methods to encode sequence clips into static image templates, such as Motion History Images (MHIs) [2], Dynamic Images (DIs) [10, 4] or Dynamic Appearance (DA) [13]. Such representations encapsulate the dynamic appearance of a sequence in a compact and visually interpretable manner that is able to encode the gradual effects of an action to the scene elements. In this work, we adopt the direction presented

in [3] and also use DIs for this purpose due to their superior representation capacity or power compared to MHIs [10].

4.2. Selecting the most informative repetitions

Not all repetitions of repetitive actions are equally informative. Ideally, we would like to model all repetition segments using a sufficient amount of sampled frames for each. However, existing HAR deep models (e.g., the Temporal Segment Network (TSN), by Wang *et al*. [28]) are able to process only for up to three concurrent input segments. Given a repetition-based segmentation of the input sequence, this would account for only three repetitions.

Under these conditions, we propose an unsupervised process for ranking the repetition segments based on information content and distinctiveness compared to the initial execution segment for each sample, which will generate a *codebook* representing the N most characteristic repetition segments. To achieve this we exploit the KKZ cluster seeding algorithm by Katsavounidis *et al*. [16], which allows for constructing a codebook, *C* of arbitrary size, containing the set of most distinct segments (*codewords*), $y_i \in R^k$. To construct the codebook, at each iteration of the KKZ algorithm, the candidate segment with the largest distance from the codebook is chosen to be the new codeword.

The first item of the codebook is always set to be the initial (first) execution of the action. The goal is to populate the codebook with the most distinctive repetitions. To compare each candidate repetition with the codewords, we use their distance as measured by the Dynamic Time Wrapping (DTW) algorithm when applied to their frame sequences. Each frame is first encoded into a $[1 \times 1024]$ feature vector using the VGG-16 network [25], pretrained on ImageNet [6]. The outcome is a ranking of the repetition segments based on content difference, which allows for a better selection of the most informative segments. As an example, Figure 1 shows four segments that were identified by the proposed KKZ-based strategy for a carrot dicing video. This leads to a better selection of the most informative repetition segments that depict the gradual effect of the dicing action on the carrot, compared to the naïve selection of the first four repetition segments.

4.3. Infusing repetition-awareness to HAR models

Early deep HAR methods [25, 9, 10] do not consider the temporal dynamics of the actions, but instead focus on appearance and short motion variations. Although such a strategy is highly effective in short-range action modeling, it proves itself insufficient for recognizing long-term actions. For long-range action modeling these methods apply temporal sampling using predefined windows, leading to a risk of "skipping" important action-related information.

To address this issue, and achieve both short- and long-range modeling of action sequences, recent state-of-the-art HAR deep models opt for configurations of the input sequence that rely on either segment-wise input sets with segment prediction consensus mechanisms [28, 35, 27], or inputs sampled at different temporal resolutions (speeds) [8, 32, 21]. In both schemes the input sequence is sampled uniformly without any prior consideration of the presence of information redundancy and the preservation of important information cues. In the previous sections we argued that the presence of a repetition-centered segmentation stage has the potential to address these issues. To prove this claim, we showcase the integration of such strategies in two exemplar, state of the art deep models for HAR. Moreover, we propose design modifications to further increase the exploitation of action repetitiveness towards increasing their performance.

4.3.1 Enhancing segment-based methods

Segment-wise video processing is perhaps the dominant input configuration approach of the recent HAR deep methods, since it expands the temporal view of the video content for the model. It was first shown to be effective by Wang *et al*. [28], with their *Temporal Segment Network (TSN)* model and was followed by numerous subsequent works that were based on their design, such as Temporal Relation Network (TRN) [35], Temporal Shift Module (TSM) [20] and Temporal Difference Network (TDN) [27].

As an exemplar method for this category, we focused on the recent Temporal Difference Network (TDN) [27]. As shown in Figure 2(a) in the rectangular box, TDN first divides a video into 8 segments uniformly along time. For each temporal segment, it randomly samples a key-frame and *T* adjacent frames. Short-term modeling operates at a segment-wise level with RGB differences between the key and neighboring frames, which is then fused with an RGB CNN representation of the segment's keyframe via a residual connection, aiming at capturing short-term motion content. Long-term modeling is achieved with a module that operates on cross-segment temporal differences with multi-scale and bidirectional attention mechanisms and aims to learn the long-range temporal structure of the action. The final action estimate is produced with an average pooling of the segment-wise representations.

To integrate the notion of repetitiveness in this model, we propose a new model variant, dubbed TDN_{RDI}, with the following modifications (depicted in Figure 2(a)).

Input configuration: The first intuitive approach is to exploit the repetition-based segmentation as a more meaningful way to define the temporal segments used as inputs to TDN. In our input configuration scheme, the starting segment in TDN is the initial execution segment. For the remaining segment spots, considering the segment processing capacity of the model (1+7 segments), we follow the repetition segment ranking approach described in Section 4.2

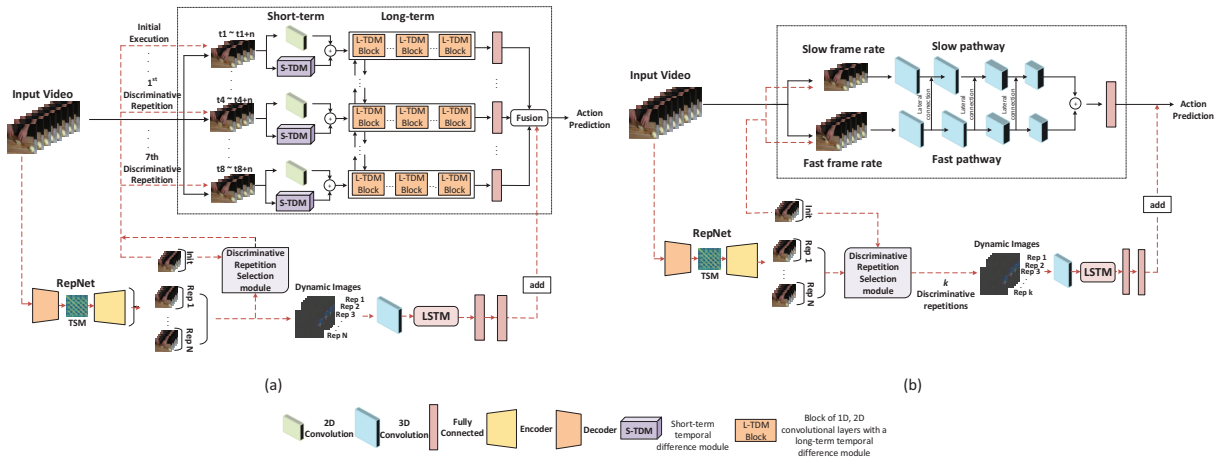


Figure 2: Original (rectangular box, with solid lines indicating original input) and proposed modifications (unbounded subnet and new inputs depicted with red dotted lines): TDN (a) and SlowFast (b) architectures. In TDN, the original architecture uses a segment-wise split of the video.

to select the seven most discriminative ones, sorted in an ascending (by repetition index) order. We note that in a sample case with less than 7 repetitions, we duplicate segments. With this repetition-centered input segment configuration policy, the short-term dynamic learning module of TDN is able to access a better-defined ordering of the action steps, and inherently formulate a stronger representation of the action. Moreover, the segmental representation difference operations in the long-term module are guided to indirectly express the fine-grained differences between each execution, which encode the effects of the repetitive action execution on the object of interest or the entire scene.

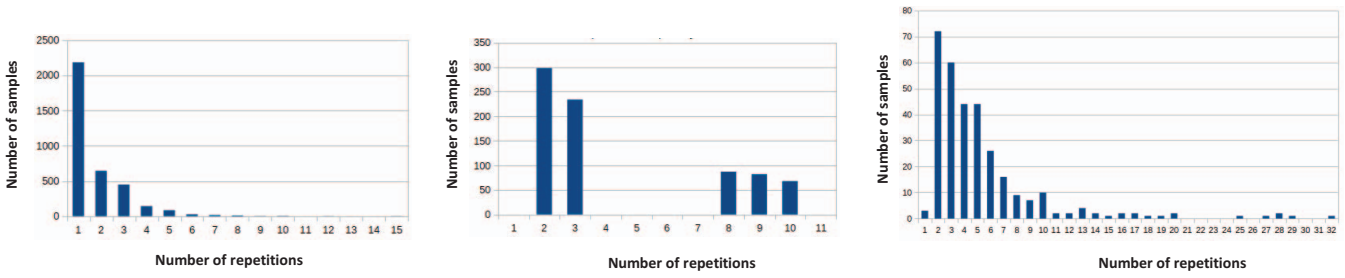
Temporal history of encoded repetition segments: The second proposed modification on TDN is the incorporation of a subnet that presents in a compact and complete way the history of the action-imposed effects on scene elements. For this, we utilized the Dynamic Image (DI) encodings [10] of all repetition segments. The set of repetition segment DIs is introduced to a two-layer (Conv3D, LSTM) network. This subnet aims to learn the spatio-temporal representation changes corresponding to the deformations on the shape/appearance of scene objects due to the repetitive action execution. As a final step, we fuse the representation of this branch with the representation of the core TDN. Such knowledge acts as a complementary information for the detailed appearance representations of the selected distinctive repetition segments that are introduced in their RGB format in the core TDN branch, and provides a compact representation of the entire action execution history.

4.3.2 Enhancing temporal multi-resolution methods

To capture the tempo and motion variations of actions while also retaining knowledge about the spatial semantics of the visual content of the action, this set of methods follow a temporal multi-resolution input sampling and processing scheme, either at a frame-level [8], or at a feature-level [32, 21]. In this work, we focus on the SlowFast model [8], shown in the rectangular box of Figure 2(b). In SlowFast, an input video is sampled at two different temporal speeds/pathways, (a) the *slow* pathway at low frame rates, to learn the semantics of the finer spatial video content, and, (b) the *fast* pathway at a high temporal resolution, to capture a range of motion speed changes. The representations from the two temporal scales are fused at various model levels with lateral connections. To transform SlowFast into a repetition-aware model, we examined two variations of the model (Figure 2(b)).

$SlowFast_{DI}$: We re-structure the *slow* pathway to sample the required frames from the DIs temporal encodings of the repetition segments (4 most discriminative), and the *fast* pathway to operate on the initial execution segment. The basis for this design choice is that the initial execution of the action usually provides a better view of the action steps at slower tempo compared to the subsequent repetitions, which will guide the model to a clearer view of the motion motif of the action. Contrary, the repetition segments will contain the action-inflicted changes on the scene elements, and thus will allow for a better representation of the action's effects at a spatial domain level. This is on par with the design purpose of the slow pathway, which is to capture the key scene appearance characteristics.

$SlowFast_{RDI}$: we re-structure the *slow* pathway to sample the required frames from the last repetition segment and the *fast* pathway to operate on the initial execution segment, following the same intuitive route as in the previous variant. The input difference on the slow pathway stems from the



(a) HMDB-51 (b) CountixHAR (c) GradEffects

Figure 3: Repetition count distributions of *HMDB51* (a), *CountixHAR* (b), and, *GradEffects* (c) datasets.

assumption that the last repetition potentially encapsulates the most noticeable action effects on scene elements, and should be considered since the action outcome is a discriminating factor when distinguishing between actions with similar motion and appearance characteristics.

As with TDN, we incorporate on SlowFast a repetition history modeling sub-net to learn the action-induced effects on the scene elements during each execution of the action. This sub-net is structured as in the TDN variant. Its input is comprised of the DIs of the K most distinct repetitions[1].

5. Experiments

Our experiments examine the impact of a repetition-aware model design under these policies:

- ***naïve***, in which each repetition segment is used as a distinct sample, without modifications on the input sampling and structure of the original models. We refer those cases as TDN$_R$ and SlowFast$_R$.
- **repetition-centric**, refers to the proposed model variants, TDN$_{RDI}$, SlowFast$_{RDI}$ & SlowFast$_{DI}$.

5.1. Datasets

We evaluate the performance of the proposed methodology on the following datasets, whose repetition count distributions are illustrated in Figure 3.

CountixHAR: It is introduced in our previous work [3] and is based on the *Countix* dataset by Dwidebi *et al*. [7]. Contains actions with 2 to 10 repetitions per sample.

HMDB-51: The *HMDB-51* dataset has a small percentage of repetitive actions (average 1.2 rep/sample).

GradEffects: is a new dataset[2] that is a subset of *Countix* [7]. The videos in GradEffects have been annotated with the temporal boundaries of the repetition segments and the repetition counts. GradEffects consists of 10 action classes, 8 of which belong to the original *Countix* dataset, (*bench pressing, front raises, jumping jacks, planning wood, sawing wood, slicing onion, push up*), and 2 additional classes (*slicing carrot, dicing carrot*). The purpose of the new action classes is to investigate the contribution of the proposed method on distinguishing between fine-grained action cases by exploiting the information content from the action-inflicted gradual effects. These two classes are typical fine-grained action cases, since they exhibit similar motion/appearance characteristics. For each action class, we annotate 40 action clips to ensure dataset balance, and apply an 80-20 train/test split scheme. The repetition counts reach up to 32 repetitions, at various durations.

[1] K in TDN is set to 7 whereas, for SlowFast is set to 4.

[2] Project & Dataset page: http://www.ics.forth.gr/cvrl/rephar/.

5.2. Training and testing configurations

For TDN we use the ResNet50-based version with 8 frames for each video, pre-trained on Kinetics-400 [17]. The batch size was set to 8, and the learning rate to 0.01, with the learning rate decay that is reported in the original paper. For the testing phase, we followed the experimental settings reported in [27] for HMDB-51 (working on split1), which we also extend for the CountixHAR and GradEffects, and report top-1, top-5 precision score under a 10-clip and 3-crop evaluation scheme.

For SlowFast, we use 32 sampled frames for the fast pathway and 4 sampled frames for the slow pathway and the reported training configurations for the Kinetics dataset in the original paper. We report the top-1, and top-5 accuracy scores under the 1-clip and center-crop testing scheme. The batch size was set to 16, and the number of epochs was maintained the same for each model and their variants. Finally, both original models and the examined variants were trained on RGB data, and the repetition-based history subnets on DIs of the selected repetition segments.

5.3. The effect of repetition awareness on accuracy

The experimental results shown in Table 1 indicate that the consideration of action repetitiveness in the design and learning schemes of HAR deep models contributes positively in their performance. Even the *naïve* exploitation of repetitions as distinct samples, leads to an increase in accuracy for both models, in almost every dataset case. This is

Prec@1,@5% for TDN & Acc@1,@5% for SlowFast, RepDI-Net			
	Datasets		
Method	**HMDB51**	**CountixHAR**	**GradEffects**
TDN [27]	47.90 / 77.97	19.80 / 41.10	44.10/100.0
TDN_R	53.44 / **84.57**	22.05 / 53.42	39.10 / 100.0
TDN_{RDI}	**54.02** / 84.38	**34.26 / 73.97**	**61.11 /100.0**
SlowFast [8]	30.85 / 64.31	25.59 / 48.54	50.00 / 98.00
SlowFast_R	37.91 / 61.90	26.78 / 55.10	53.06 / **100.0**
SlowFast_{DI}	**38.00** / 66.14	**27.97 / 61.20**	**55.20** / 96.92
SlowFast_{RDI}	36.57 / **67.69**	26.09 / 53.55	51.02 / 95.92
RepDI-Net [3]	49.61 / 74.27	56.18 / 77.89	75.16 / 93.75
RepDI-Net_{mod}	**50.97 / 79.00**	**58.71 / 82.75**	**77.25 / 93.75**

Table 1: Comparison of TDN, SlowFast, RepDI-Net variants on (a) *HMDB51*, (b) *CountixHAR*, (c) *GradEffects*.

a logical outcome since the model views a temporally constrained execution of the action with less frequent omissions of action steps or alterations of their execution order.

We can observe that TDN_{RDI} improves by a large margin the model's accuracy, especially for the datasets that contain a large set of repetitive actions. A similar positive effect on these datasets is also observed in the case of SlowFast, in which the more intuitive and simple set of modifications appears to better harness the information of the repetition-aware learning scheme. An interesting observation is that the contribution of the repetition-aware action learning differs between the two model designs, with the TDN's segmental splitting and consensus learning scheme better exploiting the more temporally bounded representation of the action steps, compared to SlowFast.

Finally, we incorporated our proposals on the repetition-aware model of our previous work [3] and observe an improvement of the model's scores by 1.36% on HMDB51, 2.53% on CountixHAR, and 2.09% on GradEffects. We used the closest training configuration to the ones used in TDN, and SlowFast, which is the case of 10-frame sequences, with random clip cropping.

5.4. Impact of repetition segmentation accuracy

The performance of the proposed repetition-aware models depend on the performance of the repetition segmentation method. Despite the large set of methods that tackle repetition detection in video data, all of them formulate the problem on the coarser task of repetition counting, and bypass the fine-grained task of the temporal localization of the repetitive segments. The selection of repetition counting as the learning objective results in artifacts, usually in the form of unrelated short-duration clips within the repetition segments. These artifacts, depending on their duration, can potentially affect the temporal encoding outcome of each segment. Nevertheless, repetition awareness is beneficial despite errors in the estimation of temporal repetition boundaries. This is manifested by the fact that the accuracy

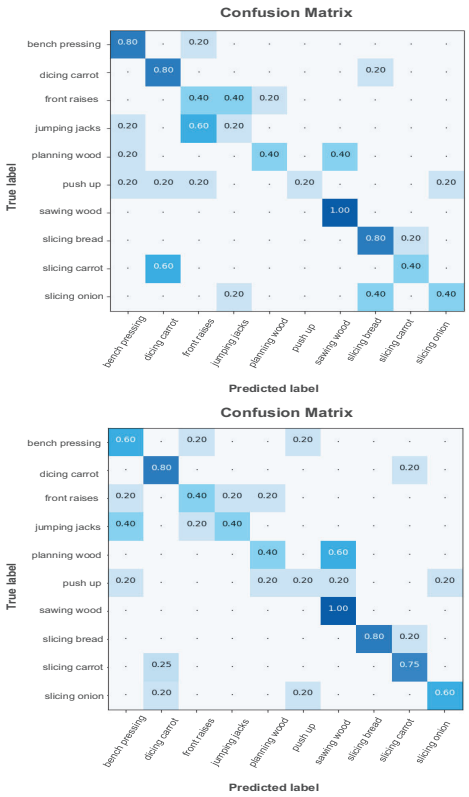


Figure 4: Confusion matrices of SlowFast[8] (top) and SlowFast_{DI} (bottom), trained on *GradEffects*.

of HAR improves (e.g., in the HMDB51 and CountixHAR datasets) regardless of the noise that is inevitably introduced by these repetition counting methods.

To better examine the impact that errors in the temporal segmentation of repetition boundaries might have in HAR in a controlled setting, we introduced Gaussian noise on the ground-truth repetition boundaries in the *GradEffects* dataset. Specifically, each repetition boundary was moved in time according to a sample of a Gaussian with μ equal to 0 and σ equal to 10% of the average duration of the repetitions. The application of the perturbed dataset on the repetition-aware model of our previous work [3], enhanced with our new modifications, leads to a 2.08% accuracy drop.

5.5. Repetition-aware recognition of action effects

To evaluate a model's ability to distinguish between fine-grained action classes by focusing on the presence of action-induced scene effects, we capitalize on the GradEffects dataset that includes the slicing carrot and dicing carrot action pair. These two actions have identical coarse motion patterns, object set, and visual features. The final state of the object affected by the action (i.e., the carrot), can be identified as the most distinctive appearance feature. Therefore, this action pair presents a challenging disambiguation

task for the model, necessitating careful attention.

The proposed scheme's ability to improve a model's performance in such action cases is evident from the confusion matrices in Figure 4, which present the classification performance of the baseline SlowFast model [8] and the proposed SlowFast$_{DI}$ variant on GradEffects. We observe that: (a) the discrimination between the two classes poses a considerable challenge, and (b) the proposed method demonstrates improved discriminatory capability for these two action classes. In a broader analysis, we find that our variant enhances performance for a majority of actions that induce a perceptible impact on the environment (e.g., slicing onion or carrot, dicing carrot), and in other cases, yields more contextually plausible misclassification outcomes. To illustrate the latter, we can examine the planning wood class, whose classification errors in the proposed scheme are attributed entirely to the sawing wood class, compared to original model that can misclassify it as bench pressing.

To further illustrate the ability of the proposed scheme to guide the model's focus on the regions of the scene in which the effect of the action takes place we employ Grad-Cam [24]. Figure 5 presents a visualization of the attention mechanisms employed by both the original SlowFast model (top row) and the SlowFast$_{DI}$ variant (bottom row) across four distinct action classes, namely, *slicing onion*, *slicing carrots*, *slicing bread*, and *jumping jacks*. Our findings reveal that SlowFast$_{DI}$ successfully directs the model's attention into regions and objects affected by the performed action, demonstrating its efficacy in guiding the model's focus towards relevant scene features. Furthermore, we observed that even for actions that do not visibly alter the surrounding environment, such as jumping jacks, the proposed scheme effectively guides the model's attention towards key scene parts that undergo changes during task execution, such as the actor's legs, waist, and hand/shoulder regions.

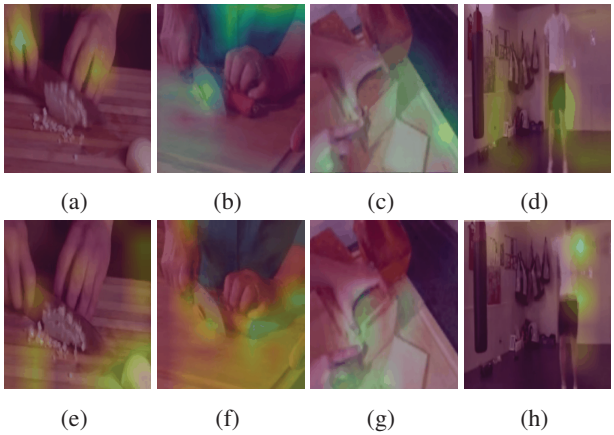


Figure 5: Grad-Cam [24] visualization of the regions of focus for SlowFast [8] (a-d) and the proposed SlowFast$_{DI}$ (e-h), for the action classes slicing onion (a,e), slicing carrot (b,f), slicing bread (c,g) and jumping jacks (d,h).

6. Discussion: *Handling non-repetitive actions*

One crucial inquiry pertains to the generalizability of the proposed learning scheme for both repetitive and non-repetitive actions. It should be noted that not all instances of an action necessarily exhibit repetitiveness, even if the action itself is generally categorized as repetitive. Moreover, not all actions can be repetitive. HAR models should possess the capability to effectively leverage repetitiveness as a means of modeling actions whenever it is present.

In non-repetitive action instances, the repetition segmentation module can yield two possible outcomes: either erroneous estimations or failure to detect repetition segments. In the former scenario, the proposed repetition history subnet encodes the progress of the action, effectively temporally encoding and modeling segmented portions of the action. This provides a supplementary temporal representation of the sample. In the latter case, the repetition counting method will indicate a repetition count of zero, which, according to the proposed learning scheme, will result in the initial sequence being sampled using the input processing strategies of the initial models. In this situation, the proposed repetition history subnet processes the DI encodings of K uniformly sampled segments from the entire input sequence, thus, functioning again as a temporal modeling module, complementary to the encoded initial input.

7. Summary

This paper introduced effective mechanisms for incorporating a repetition-aware input segmentation step in the learning of deep models for HAR. In addition, we proposed a set of intuitive modifications of two state of the art HAR models that extend the exploitation capabilities of action repetitiveness, further improving the performance of these models. A series of experiments showed the significant benefits of the proposed approach. Our investigation demonstrates that considering action repetitiveness in the design and learning process of HAR models has a significant positive impact on HAR accuracy. Future research will exploit spatiotemporal (rather than purely temporal) localization of repetitive actions as a means of focusing on the spatiotemporal volume at which repetitive actions occur towards better capturing action effects on the involved objects.

Acknowledgments

This research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) 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.

References

[1] Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *Acm Computing Surveys (Csur)*, 43(3):1–43, 2011.

[2] Md Atiqur Rahman Ahad, Joo Kooi Tan, Hyoungseop Kim, and Seiji Ishikawa. Motion history image: its variants and applications. *Machine Vision and Applications*, 23(2):255–281, 2012.

[3] Konstantinos Bacharidis and Antonis Argyros. Exploiting the nature of repetitive actions for their effective and efficient recognition. *Frontiers in Computer Science*, 4, 2022.

[4] Hakan Bilen, Basura Fernando, Efstratios Gavves, and Andrea Vedaldi. Action recognition with dynamic image networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2799–2813, 2017.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10387–10396, 2020.

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[10] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5378–5387, 2015.

[11] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and vision computing*, 60:4–21, 2017.

[12] Huazhang Hu, Sixun Dong, Yiqun Zhao, Dongze Lian, Zhengxin Li, and Shenghua Gao. Transrac: Encoding multi-scale temporal correlation with transformers for repetitive action counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19013–19022, 2022.

[13] Guoxi Huang and Adrian G. Bors. Dynamic appearance: A video representation for action recognition with joint training, 2022.

[14] Amlan Kar, Nishant Rai, Karan Sikka, and Gaurav Sharma. Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3376–3385, 2017.

[15] Giorgos Karvounas, Iason Oikonomidis, and Antonis Argyros. Reactnet: Temporal localization of repetitive activities in real-world videos. *arXiv preprint arXiv:1910.06096*, 2019.

[16] Ioannis Katsavounidis, C.-C. Jay Kuo, and Zhen Zhang. A new initialization technique for generalized lloyd iteration. *Signal Processing Letters, IEEE*, 1:144 – 146, 11 1994.

[17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[18] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6232–6242, 2019.

[19] Ofir Levy and Lior Wolf. Live repetition counting. In *Proceedings of the IEEE international conference on computer vision*, pages 3020–3028, 2015.

[20] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[21] Yuanzhong Liu, Zhigang Tu, Hongyan Li, Chi Chen, Baoxin Li, and Junsong Yuan. Slow-fast visual tempo learning for video-based action recognition. *arXiv preprint arXiv:2202.12116*, 2022.

[22] Costas Panagiotakis, Giorgos Karvounas, and Antonis Argyros. Unsupervised detection of periodic segments in videos. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 923–927. IEEE, 2018.

[23] Tom FH Runia, Cees GM Snoek, and Arnold WM Smeulders. Repetition estimation. *International Journal of Computer Vision*, 127(9):1361–1383, 2019.

[24] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Action recognition in video sequences using deep bi-directional lstm with cnn features. *IEEE Access*, 6:1155–1166, 2018.

[27] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1895–1904, 2021.

[28] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[29] Xianyuan Wang, Zhenjiang Miao, Ruyi Zhang, and Shanshan Hao. I3d-lstm: A new model for human action recognition. *IOP Conference Series: Materials Science and Engineering*, 569(3):032035, jul 2019.

[30] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6026–6035, 2018.

[31] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1278–1287, 2019.

[32] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 591–600, 2020.

[33] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[34] Shiwen Zhang, Sheng Guo, Weilin Huang, Matthew R Scott, and Limin Wang. V4d: 4d convolutional neural networks for video-level representation learning. *arXiv preprint arXiv:2002.07442*, 2020.

[35] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 803–818, 2018.

[36] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*, 2020.