signals

MDPI

Article

HalluciNet-*ing* Spatiotemporal Representations Using a 2D-CNN

Paritosh Parmar [1,*] and Brendan Morris [2]

[1] Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[2] Department of Electrical & Computer Engineering, University of Nevada, Las Vegas, NV 89119, USA; brendanmorris@unlv.nevada.edu
* Correspondence: parmap1@unlv.nevada.edu

Abstract: Spatiotemporal representations learned using 3D convolutional neural networks (CNN) are currently used in state-of-the-art approaches for action-related tasks. However, 3D-CNN are notorious for being memory and compute resource intensive as compared with more simple 2D-CNN architectures. We propose to hallucinate spatiotemporal representations from a 3D-CNN teacher with a 2D-CNN student. By requiring the 2D-CNN to predict the future and intuit upcoming activity, it is encouraged to gain a deeper understanding of actions and how they evolve. The hallucination task is treated as an auxiliary task, which can be used with any other action-related task in a multitask learning setting. Thorough experimental evaluation, it is shown that the hallucination task indeed helps improve performance on action recognition, action quality assessment, and dynamic scene recognition tasks. From a practical standpoint, being able to hallucinate spatiotemporal representations without an actual 3D-CNN can enable deployment in resource-constrained scenarios, such as with limited computing power and/or lower bandwidth. We also observed that our hallucination task has utility not only during the training phase, but also during the pre-training phase.

Keywords: action recognition; scene recognition; action quality assessment; activity recognition; deep learning; computer vision; convolutional neural networks; multitask learning; transfer learning

Citation: Parmar, P.; Morris, B. HalluciNet-*ing* Spatiotemporal Representations Using a 2D-CNN. *Signals* **2021**, *2*, 604–618. https://doi.org/10.3390/signals2030037

Academic Editor: Richard J. Povinelli

Received: 25 May 2021
Accepted: 17 August 2021
Published: 8 September 2021

1. Introduction

Spatiotemporal representations are densely packed with information regarding both the appearance and salient motion patterns occurring in the video clips, as illustrated in Figure 1. Due to this representational power, they are currently the best performing models on action-related tasks, such as action recognition [1–4], action quality assessment [5–9], skills assessment [10], and action detection [11]. This representation power comes at the cost of increased computational complexity [12–15], which makes 3D-CNNs unsuitable for deployment in resource-constrained scenarios.

The power of 3D-CNNs comes from their ability to attend to the salient motion patterns of a particular action class. In contrast, 2D-CNNs are generally used for learning and extracting spatial features pertaining to a single frame/image; thus, by design, they do not take into account any motion information and, therefore, lack temporal representation power. Some works [16–19] have addressed this by using optical flow, which will respond at all pixels that have moved/changed. This means the optical flow can respond to cues both from the foreground motion of interest, as well as the irrelevant activity happening in the background. This background response might not be desirable since CNNs have been shown to find *short cuts* to recognize actions not from the meaningful foreground, but from background cues [20,21]. These kinds of short cuts might still be beneficial for action recognition tasks but not in a meaningful way, that is, the 2D network is not actually learning to understand the action itself, but rather the contextual cues and clues. Despite these shortcomings, 2D-CNNs are computationally lightweight, which makes them suitable for deployment on edge devices.

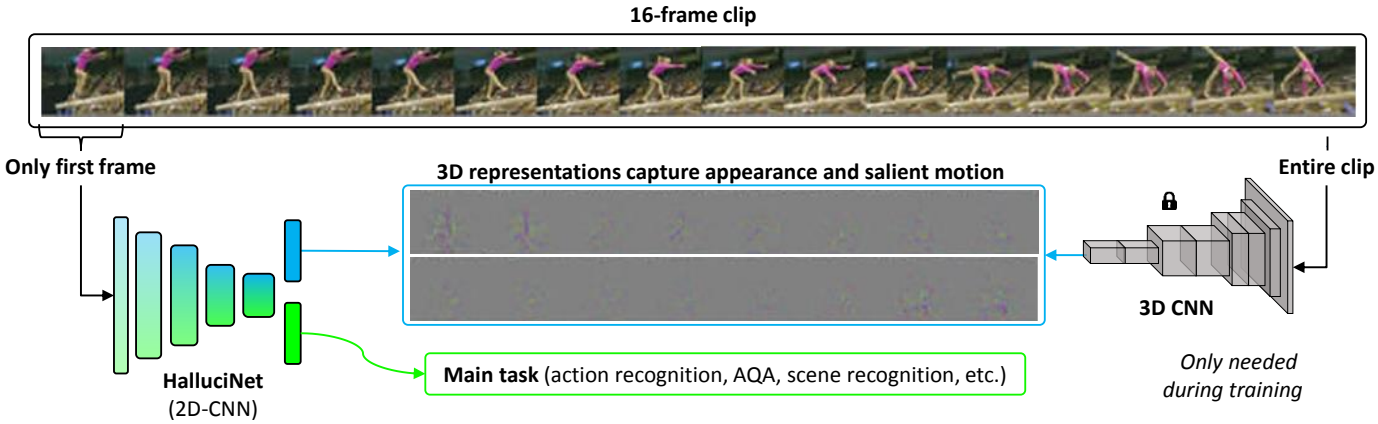

Figure 1. Multitask learning with HalluciNet. HalluciNet (2D-CNN) is jointly optimized for the main task and to hallucinate spatiotemporal features (computed by an actual 3D-CNN) from a single frame.

In short, 2D-CNNs have the advantage of being computationally less expensive, while 3D-CNNs extract spatiotemporal features that have more representation power. In our work, we propose a way to combine the best of both worlds—rich spatiotemporal representation with low computational cost. Our inspiration comes from the observation that given even a single image of a scene, humans can predict how the scene might evolve. We are able to do so because of our experience and interaction in the world, which provides a general understanding of how other people are expected to behave and how objects can move or be manipulated. We propose to hallucinate spatiotemporal representations as computed by a 3D-CNN, using a 2D-CNN, utilizing only a single still frame (see Figure 1). The idea is to force a 2D-CNN to predict the motion that will occur in the next frames, without ever having to actually see it.

Contributions: We propose a novel multitask approach, which incorporates an auxiliary task of approximating 3D-CNN representations using a 2D-CNN and a single image. It has the following benefits:

- Conceptually, our hallucination task can provide a richer, stronger supervisory signal that can help the 2D-CNN to gain a deeper understanding of actions and how a given scene evolves with time. Experimentally, we found our approach to be beneficial in the following computer vision tasks:
  1. Action recognition (actions with short- and long-term temporal dynamics).
  2. Action quality assessment.
  3. Scene recognition.

  Furthermore, we also found hallucination task to be useful during the following:
  1. The pretraining phase.
  2. The training phase.
- Practically, approximating spatiotemporal features, instead of actually computing them, is useful for the following:
  1. Limited compute power (smart video camera systems, lower-end phones, or IoT devices).
  2. Limited/expensive bandwidth (Video Analytics Software as a Service (VA SaaS)), where our method can help reduce the transmission load by a factor of 15 (need to transmit only 1 frame out of 16).

Many computer vision efforts in areas such as automated (remote) physiotherapy (action quality assessment), which are targeted for low-income groups, make use of 3D-CNNs. It is more likely that a low income demographic would have devices with low computational resources and restricted communication resources, which are not suitable to run 3D-CNNs; in these cases, we can just hallucinate spatiotemporal representations, instead of using actual 3D-CNNs and a large number of frames.

2. Related Work

Our work is related to predicting features, developing efficient/light-weight spatiotemporal network approaches, and distilling knowledge. Next, we briefly compare and contrast our approach to the most closely related works in the literature.

Capturing information in future frames: Many works have focused on capturing information in future frames [16–18,22–30]. Generating future frames is a difficult and complicated task, and usually requires the disentangling of background, foreground, low-level and high-level details, and modeling them separately. Our approach to predicting features is much simpler. Moreover, our goal is not to a predict a pixel-perfect future, but rather to make predictions at the semantic level.

Instead of explicitly generating future frames, works such as [16–19] focused on learning to predict the optical flow (very short-term motion information). These approaches, by design, require the use of an encoder and a decoder. Our approach does not require a decoder, which reduces the computational load. Moreover, our approach learns to hallucinate features corresponding to 16 frames, as compared to motion information in two frames. Experiments confirm the benefits of our method over optical flow prediction.

Bilen et al. [29] introduced a novel, compact representation of a video called a "dynamic image", which can be thought of as a summary of full videos in a single image. However, computing a dynamic image requires access to all the corresponding frames, whereas HalluciNet requires processing just a single image.

Predicting features: Other works [27,31,32] proposed predicting features. Our work is closest to [32], where the authors proposed hallucinating depth using the RGB input, whereas we propose hallucinating the spatiotemporal information. Reasoning about depth information is different from reasoning about spatiotemporal evolution.

Efficient Spatiotemporal Feature Computation: Numerous works have developed approaches to make video processing more efficient, either by reducing the required input evidence [33–36], or explicitly, through more efficient processing [12,13,37–42].

While these works aim to address either reducing the visual evidence or developing a more efficient architecture design, our solution to hallucinate (without explicitly computing) spatiotemporal representations using a 2D-CNN from a single image aims to solve both the problems, while also providing stronger supervision. In fact, our approach, which focuses on improving the backbone CNN, is complementary to some of these developments [42,43].

3. Best of Both Worlds

Since humans are able to predict future activity and behavior through years of experience and a general understanding of "how the world works", we would like to develop a network that can understand an action in a similar manner. To this end, we propose a teacher–student network architecture that asks a 2D-CNN to use a single frame to hallucinate (predict) 3D features pertaining to 16 frames.

Let us consider the example of a gymnast performing her routine as shown in Figure 1. In order to complete the hallucination task, the 2D-CNN should do the following:

- Learn to identify that there's an actor in the scene and localize her;
- Spatially segment the actors and objects;
- Identify that the event is a balance beam gymnastic event and the actor is a gymnast;
- Identify that the gymnast is to attempt a cartwheel;
- Predict how she will be moving while attempting the cartwheel;
- Approximate the final position of the gymnast after 16 frames, etc.

The challenge is understanding all the rich semantic details of the action from only a single frame.

3.1. Hallucination Task

The hallucination task can be seen as distilling knowledge from a better teacher network (3D-CNN), f_t, to a lighter student network (2D-CNN), f_s. The teacher, f_t, is

pretrained and kept frozen, while the parameters of the student, f_s, are learned. Mid-level representations can be computed as follows:

$$\phi_t = f_t(F_0, F_1, ..., F_{T-1}) \quad (1)$$

$$\phi_s = f_s(F_0) \quad (2)$$

where F_T is the T-th video frame.

The hallucination loss, $\mathcal{L}_{hallu}$ encourages f_s to regress ϕ_s to ϕ_t by minimizing the Euclidean distance between ϕ_s and ϕ_t:

$$\mathcal{L}_{hallu} = |\sigma(\phi_s) - \sigma(\phi_t)|^2. \quad (3)$$

Multitask learning (MTL):

Reducing computational cost with the hallucination task is not the only goal. Since the primary objective is to better understand activities and improve performance, hallucination is meant to be an auxiliary task to support the main action-related task (e.g., action recognition). The main task loss (e.g., classification loss), $\mathcal{L}_{mt}$, is used in conjunction with the following hallucination loss:

$$\mathcal{L}_{MTL} = \mathcal{L}_{mt} + \lambda \mathcal{L}_{hallu} \quad (4)$$

where λ is a loss balancing factor. The realization of our approach is straightforward, as presented in Figure 1.

3.2. Stronger Supervision

In a typical action recognition task, a network is only provided with the action class label. This may be considered a weak supervision signal since it provides a single high-level semantic interpretation of a clip filled with complex changes. More dense labels at lower semantic levels are expected to provide stronger supervisory signals, which could improve action understanding.

In this vein, joint actor–action segmentation is an actively pursed research direction [44–48]. Joint actor–action segmentation datasets [49] provide detailed annotations, through significant annotation efforts. In contrast, our spatiotemporal hallucination task provides detailed supervision of a similar type (though not exactly the same) for free. Since 3D-CNN representations tend to focus on actors and objects, 2D-CNN can develop a better general understanding about actions through actor/object manipulation. Additionally, the 2D representation is less likely to take shortcuts—ignoring the actual actor and action being performed, and instead doing recognition based on the background [20,21]—as it cannot hallucinate spatiotemporal features, which mainly pertain to the actors/foreground from the background.

3.3. Prediction Ambiguities

In general, the prediction of future activity with a single frame could be ambiguous (e.g., opening vs. closing a door). However, a study has shown that humans are able to accurately predict immediate future action from a still image 85% of the time [27]. So, while there may be ambiguous cases, there are many other instances where causal relationships exist and the hallucination task can be exploited. Additionally, low-level motion cues can be used to resolve ambiguity (Section 4.4).

4. Experiments

We hypothesize that incorporating the hallucination task is beneficial by providing a deeper understanding of actions. We evaluate the effect of incorporating the hallucination task in the following settings:

- (Section 4.1) Actions with short-term temporal dynamics.
- (Section 4.2) Actions with long-term temporal dynamics.

- (Section 4.3) Non-action task.
- (Section 4.4) Hallucinating from two frames, instead of a single frame.
- (Section 4.5) Effect of injecting hallucination task during pretraining.

Choice of networks: In principle, any 2D- or 3D-CNNs could be used as student or teacher networks, respectively. Noting the SOTA performance of 3D-ResNeXt-101 [3] on action recognition, we choose to use it as our teacher network. We considered various student models. Unless otherwise mentioned, our student model is VGG11-bn, and pretrained on the ImageNet dataset [50]; the teacher network was trained on UCF-101 [51] and kept frozen. We named the 2D-CNN trained with the side-task hallucination loss as HalluciNet, and the one without hallucination loss as (vanilla) 2D-CNN, while the $\text{HalluciNet}_{\text{direct}}$ variant, which directly uses hallucinated features for the main action recognition task.

Which layer to hallucinate? We chose to hallucinate the activations of the last bottleneck group of 3D-ResNeXt-101, which are 2048-dimensional. Representations of shallower layers will have higher dimensionality and will be less semantically mapped.

Implementation details: We used PyTorch [52] to implement all of the networks. Network parameters were optimized using an Adam optimizer [53] with a beginning learning rate of 0.0001. λ in Equation (4) was set to 50, unless specified otherwise. Further experiment specific details are presented with the experiment. The codebase will be made publicly available.

Performance baselines: Our performance baseline was a 2D-CNN with the same architecture, but was trained without hallucination loss (vanilla 2D-CNN). In addition, we also compared the performance against other popular approaches from the literature, specified in each experiment.

4.1. Actions with Short-Term Temporal Dynamics

In the first experiment, we tested the influence of the hallucination task for general action recognition. We compared the performance with two single frame prediction techniques: dense optical flow prediction from a static image [17], and motion prediction from a static image [19].

Datasets: The following action recognition datasets were considered.

1. UCF101 [51] is an action recognition dataset of realistic in-the-wild action videos, collected from YouTube, having 101 action categories. With 13,320 videos from 101 action categories, UCF101 provides the largest diversity in terms of actions and with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc., and it is the most challenging dataset to date. The action categories can be divided into five types: (1) human–object interaction; (2) body motion only; (3) human–human interaction; (4) playing musical instruments; and (5) sports.
2. HMDB-51 [54] is collected from various sources, mostly from movies, and a small proportion from public databases such as the Prelinger archive, YouTube and Google videos. This dataset contains 6849 clips divided into 51 action categories, each containing a minimum of 101 clips. The action categories can be grouped in five types: (1) general facial actions; (2) facial actions with object manipulation; (3) general body movements; (4) body movements with object interaction; and (5) body movements for human interaction. Since HMDB-51 video sequences are extracted from commercial movies as well as YouTube, it represents a fine multifariousness of light conditions, situations and surroundings in which the action can appear, captured with different camera types and recording techniques, such as points of view.

In order to be consistent with the literature, we adopted their experimental protocols. Center frames from the training and testing samples were used for reporting performance, and are named UCF- and HMDB-static, as in the literature [19].

Metric: We report the top-1 frame/clip-level accuracy (in %).

We summarize the performance on the action recognition task in Table 1a. We found that on both datasets, incorporating the hallucination task helped. Our HalluciNet outperformed prior approaches [17,19] on both the UCF-101 and HMDB-51 datasets. Moreover, our method has an advantage of being computationally lighter than [19], as it does not use a flow image generator network. Qualitative results are shown in Figure 2. In the successes, ambiguities were resolved. The failure cases tended to confuse semantically similar classes with similar motions, such as FloorGymnastics/BalanceBeam or Kayaking/Rowing. To evaluate the quality of the hallucinated representations themselves, we directly used those representations for the main action recognition task (HalluciNet$_{\text{direct}}$). We noticed that the hallucinated features had strong performance, improved on the 2D-CNN, and, in fact, performed best on the UCF-static.

Table 1. (**a**) Action recognition results and comparison. (**b**) HalluciNet helps recent developments, such as TRN and TSM. (**c**) Multiframe inference on better base model. (**b**,**c**) Evaluated on UCF101.

(**a**)		
Method	**UCF-Static**	**HMDB-Static**
App stream [19]	63.60	35.10
App stream ensemble [19]	64.00	35.50
Motion stream [19]	24.10	13.90
Motion stream [17]	14.30	04.96
App + Motion [19]	65.50	37.10
App + Motion [17]	64.50	35.90
Ours 2D-CNN	64.97	34.23
Ours HalluciNet	69.81	**40.32**
Ours HalluciNet$_{\text{direct}}$	**70.53**	39.42
(**b**)		
Method	**Accuracy**	
TRN (R18)	69.57	
TRN (HalluciNet(R18))	**69.94**	
TSM (R18)	71.40	
TSM (HalluciNet(R18))	**73.12**	
(**c**)		
Model	**Accuracy**	
Resnet-50	76.74	
HalluciNet(R50)	**79.83**	

Next, we used the hallucination task to improve the performance of recent developments, TRN [43] and TSM [42]. We used Resnet-18 (R18) as the backbone for both, and implemented single, center segment, 4-frame versions of both. For TRN, we considered the multiscale version. For TSM, we considered the online version, which is intended for real-time processing. For both, we sampled 4 frames from the center 16 frames. We used $\lambda = 200$. Their vanilla versions served as our baselines. The performance on UCF101 is shown in Table 1b.

We also experimented with a larger, better base model, Resnet-50. In this experiment, we trained using all the frames, and not only the center frame; during testing, we averaged the results over 25 frames. We used $\lambda = 350$. Results on UCF101 are shown in Table 1c.

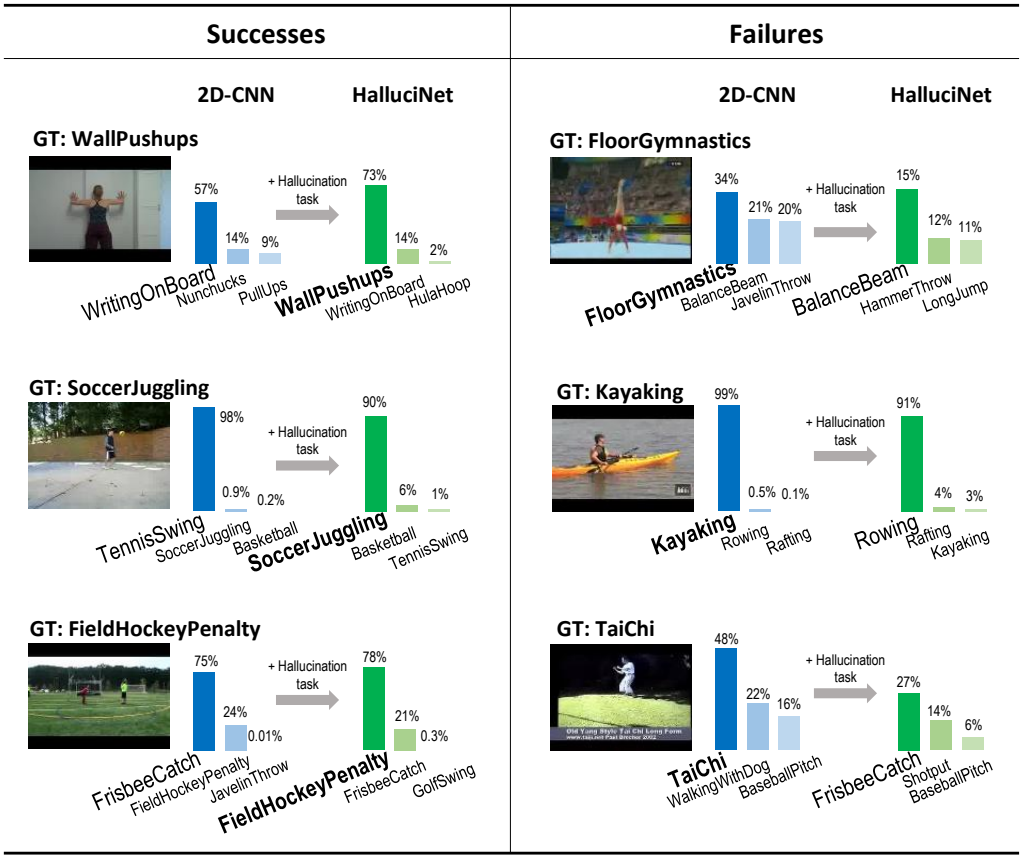

Figure 2. Qualitative results. The hallucination task helps improve performance when the action sample is visually similar to other action classes, and a motion cue is needed to distinguish them. However, sometimes, HalluciNet makes incorrect predictions when the motion cue is similar to that of other actions, and dominates over the visual cue. Please zoom in for a better view.

Finally, in Table 2, we compare our predicted spatiotemporal representation, HalluciNet$_{\text{direct}}$, and the actual 3D-CNN. Hallucinet improved upon the vanilla 2D-CNN, though well below the actual 3D-CNN. However, the performance trade-off resulted in only 6% of the computational cost of the full 3D-CNN. We also observed a reduction in the data needed to be transmitted. Here, our 3D-CNN used 112 $\times$ 112 pixel frames as input (mean image size: 3.5 KB), while our 2D-CNN used 224 $\times$ 224 pixels (mean image size: 13.94 KB) as input. In this case, we observed a reduction of more than 4 times. In cases where 3D-CNNs use a larger resolution input, we can expect a much larger reduction. For example, if the 3D-CNN uses frames of resolution of 224 $\times$ 224 pixels, then the data transmission would be reduced by 16 times.

Table 2. Cost vs. accuracy comparison. We measure the times on a Titan-X GPU.

Model	Accuracy		Time/Inf.	FLOPs/Param	Data Transmitted
	UCF	HMDB			
Ours 2D-CNN	64.97	34.23	3.54 ms	58	13.94 KB
Ours HalluciNet	70.53	39.42			
Ours actual 3D-CNN	87.95	60.44	58.99 ms	143	56.00 KB

4.2. Actions with Long-Term Temporal Dynamics

Although we proposed hallucinating the short-term future (16 frames), frequently, actions with longer temporal dynamics must be considered. To evaluate the utility of short-term hallucination in actions with longer temporal dynamics, we considered the task

of recognizing dives and assessing their quality. Short clips were aggregated over longer videos using an LSTM, as shown in Figure 3.

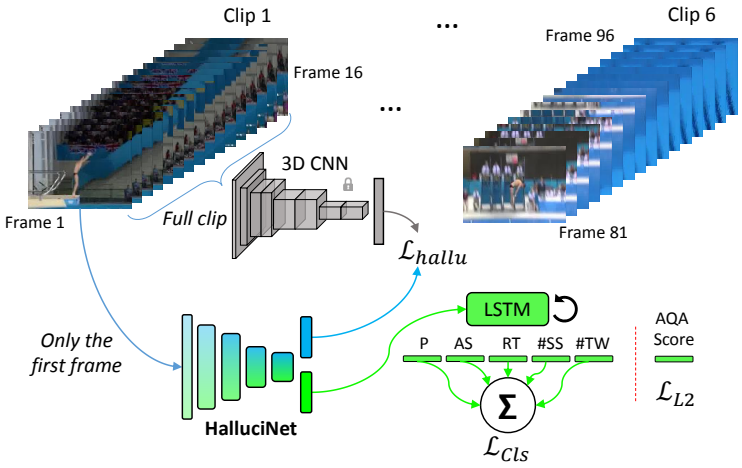

Figure 3. Detailed action recognition and action quality assessment models.

4.2.1. Dive Recognition

Task description: In Olympic diving, athletes attempt many different types of dives. In a general action recognition dataset, such as UCF101, all of these dives are grouped under a single action class: diving. However, these dives are different from each other in subtle ways. Each dive has the following five components: (a) position (legs straight or bent); (b) starting from arm stand or not; (c) rotation direction (backwards, forwards, etc.); (d) number of times the diver somersaulted in air; and (e) number of times the diver twisted in air. Different combinations of these components produce a unique type of dive (dive number). The dive recognition task comprises predicting all five components of a dive using very few frames.

Why is this task more challenging? Unlike general action recognition datasets, e.g., UCF-101 or kinetics [2], the cues needed to identify the specific dive are distributed across the entire action sequence. In order to correctly predict the dive, the whole action sequence needs to be seen. To make the dive classification task more suitable for our HalluciNet framework, we asked the network to classify a dive correctly using only a few regularly spaced frames. In particular, we truncated a diving video to 96 frames and showed the student network every 16th frame, for a total of 6 frames. Note that we are not asking our student network to hallucinate the entire dive sequence; rather, the student network is required to hallucinate the short-term future in order to "fill the holes" in the visual input datastream.

Dataset: The recently released diving dataset MTL-AQA [6], which has 1059 training and 353 test samples, was used for this task. The average sequence length is 3.84 s. Diving videos are real-world footage collected from various FINA events. The side view is used for all the videos. Backgrounds and the clothing of athletes vary.

Model: We pretrained both our 3D-CNN teacher and 2D-CNN student on UCF-101. Then, the student network was trained to classify dives. Since we would be gathering evidence over six frames, we made use of an LSTM [55] for aggregation. The LSTM was single-layered with a hidden state of 256D. The LSTM's hidden state at the last time step was passed through separate linear classification layers, one for each of the properties of a dive. The full model is illustrated in Figure 3. The student network was trained end-to-end for 20 epochs using an Adam solver with a constant learning rate of 0.0001. We also considered HalluciNet based on Resnet-18 ($\lambda = 400$). We did not consider R50 because it is much larger, compared to the dataset size.

The results are summarized in Table 3a, where we also compare them with other state-of-the-art 3D-CNN-based approaches [6,56]. Compared with the 2D baseline, Hallucinet performed better on 3 out of 5 tasks. The position task (legs straight or bent) could be

equally identifiable from a single image or clip, but the number of twists, somersaults, or direction of rotation are more challenging without seeing motion. In contrast, HalluciNet could predict the motion. Our HalluciNet even outperformed 3D-CNN-based approaches that use more frames (MSCADC [6] and Nibali et al. [56]). C3D-AVG outperformed HalluciNet, but is computationally expensive and uses 16× more frames.

Table 3. (**a**) Performance (Accuracy in % reported) comparison on dive recognition task. #Frames represents the number of frames that the corresponding method sees. P, AS, RT, SS, TW stand for position, arm stand, rotation type, number of somersaults, and number of twists. (**b**) Performance (Spearman's rank correlation in % reported) on AQA task.

(**a**)

Method	**CNN**	**#Frames**	**P**	**A**	**RT**	**SS**	**TW**
C3D-AVG [6]	3D	96	96.32	99.72	97.45	96.88	93.20
MSCADC [6]	3D	16	78.47	97.45	84.70	76.20	82.72
Nibali et al. [56]	3D	16	74.79	98.30	78.75	77.34	79.89
Ours VGG11	**2D**	**6**	**90.08**	99.43	92.07	83.00	86.69
HalluciNet (VGG11)	**2D**	**6**	89.52	99.43	**96.32**	**86.12**	**88.10**
HalluciNet (R18)	**2D**	**6**	**91.78**	99.43	95.47	**88.10**	**89.24**

(**b**)

Method	**CNN**	**#Frames**	**Corr.**
Pose+DCT [57]	-	96	26.82
C3D-SVR [5]	3D	96	77.16
C3D-LSTM [5]	3D	96	84.89
C3D-AVG-STL [6]	3D	96	89.60
MSCADC-STL [6]	3D	16	84.72
Ours VGG11	**2D**	**6**	80.39
Ours HalluciNet (VGG11)	**2D**	**6**	**82.70**
Ours HalluciNet (R18)	**2D**	**6**	**83.51**

4.2.2. Dive Quality Assessment

Action quality assessment (AQA) is another task that can highlight the utility of hallucinating spatiotemporal representations from still images, using a 2D-CNN. In AQA, the task is to measure, or quantify, *how well* an action was performed. A good example of AQA is that of judging Olympic events, such as diving, gymnastics, figure skating, etc. Like the dive recognition task, in order to correctly assess the quality of a dive, the entire dive sequence needs to be seen/processed.

Dataset: MTL-AQA [6], the same as in Section 4.2.1.

Metric: Consistent with the literature, we report Spearman's rank correlation (in %).

We followed the same training procedure as in Section 4.2.1, except that for the AQA task, we used L2 loss to train, as it is a regression task. We trained for 20 epochs with Adam as a solver and annealed the learning rate by a factor of 10 every 5 epochs. We also considered HalluciNet based on R18 ($\lambda = 250$).

The AQA results are presented in Table 3b. Incorporating the hallucination task helped improve AQA performance. Our HalluciNet outperformed C3D-SVR and was quite close to C3D-LSTM and MSCADC, although it saw 90 and 10 fewer frames, respectively. Although it does not match C3D-AVG-STL, HalluciNet requires significantly less computation.

4.3. Dynamic Scene Recognition

Dataset: Feichtenhofer et al. introduced the YUP++ dataset [58] for the task of dynamic scene recognition. It has a total of 20 scene classes. Samples from this dataset encompass a wide range of conditions, including those arising from natural within scene category

differences, seasonal and diurnal variations as well as viewing parameters. For each scene class in the dataset, there are 60 color videos, with no two samples for a given class taken from the same physical scene. Half of the videos within each class were acquired with a static camera and half were acquired with a moving camera, with camera motions encompassing pan, tilt, zoom and jitter.

The use of this dataset to evaluate the utility of inferred motion was suggested in [19]. In the work by Feichtenhofer, 10% of the samples were used for training, while the remaining 90% of the samples were used for testing purposes. Gao et al. [19] formed their own split, called *static-YUP++*.

Protocol: For training and testing purposes, we considered the central frame of each sample.

The first experiment considered standard dynamic scene recognition using splits from the literature and compared them with a spatiotemporal energy based approach (BoSE), slow feature analysis (SFA) approach, and temporal CNN (T-CNN). Additionally, we also considered versions based on Resnet50 and predictions averaged over 25 frames. As shown in Table 4a, HalluciNet showed minor improvement over the baseline 2D-CNN and outperformed studies in the literature. T-CNN might be the closest for comparison because it uses a stack of 10 optical flow frames; however, our HalluciNet outperformed it by a large margin. Note that we did not train our 3D-CNN on the scene recognition dataset/task, and used a 3D-CNN trained on the action recognition dataset, but we still observed improvements.

Table 4. (**a**) Dynamic scene recognition on YUP++. (**b**) Dynamic scene recognition on static-YUP++.

(**a**)	
Method	**Accuracy**
SFA [59]	56.90
BoSE [60]	77.00
T-CNN [61]	50.60
(Our Singleframe inference)	
Our VGG11	77.50
Our HalluciNet(VGG11)	**78.15**
(Our Multiframe inference with better base models)	
Our Resnet-50	83.43
Our HalluciNet(Resnet-50)	**84.44**
(**b**)	
Method	**Accuracy**
Appearance [19]	74.30
GT Motion [19]	55.50
Inferred Motion [19]	30.00
Appearance ensemble [19]	75.20
Appearance + Inferred Motion [19]	78.20
Appearance + GT Motion [19]	79.60
Our 2D-CNN	72.04
Our HalluciNet	**81.53**

The second experiment compared our approach with [19] in which we used their split for static-YUP++ (Table 4b). In this case, our vanilla 2D-CNN did not outperform studies in the literature, but our HalluciNet did—even when groundtruth motion information was used by im2flow [19].

4.4. Using Multiple Frames to Hallucinate

As previously discussed, there are situations (e.g., door open/close) where a single image cannot be reliably used for hallucination. However, motions cues coming from multiple frames can be used to resolve ambiguities.

We modified the single frame HalluciNet architecture to accept multiple frames, as shown in Figure 4. We processed frame F_j and frame F_{j+k} ($k > 0$) with our student 2D-CNN. In order to tease out low-level motion cues, we did ordered concatenation of the intermediate representations, corresponding to frames F_j and F_{j+k}. The concatenated student representation in the 2-frame case is as follows:

$$\phi_s = concat_\phi(\phi_s^j, \phi_s^{j+k}) \tag{5}$$

where ϕ_s^l is the student representation from frame F_l as in Equation (2). This basic approach can be extended to more frames, as well as multi-scale cases. Hallucination loss remains as a single frame case (Equation (3).

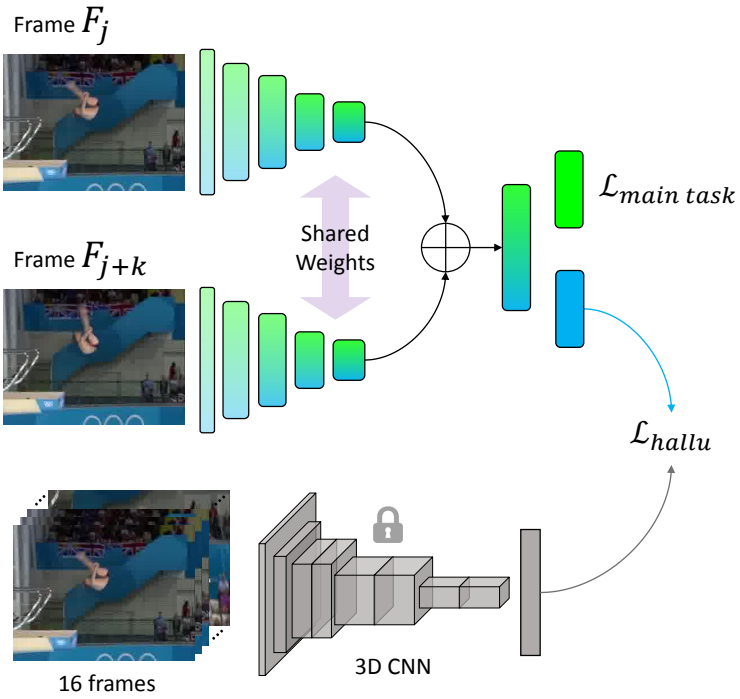

Figure 4. Multiframe architecture. Instead of using a single frame to hallucinate, representations of ordered frames are concatenated ($\oplus$), which is then used for hallucinating. Everything else remains the same as our single frame model.

In order to see the effect of using multiple frames, we considered the following two cases:

1. Single-frame baseline (HalluciNet(1f)). We set $k = 0$, which is equivalent to our standard single frame case;
2. Two-frame baseline (HalluciNet(2f)). We set $k = 3$, to give the student network f_s access to pixel changes in order to tease out low-level motion cues.

We trained the networks for both the cases, using the exact same procedure and parameters as in the single frame case, and observed the hallucination loss, $\mathcal{L}_{hallu}$, on the test set. We experimented with both kinds of actions—with short-term and long-term dynamics.

Results for short-term actions are presented in Table 5a for UCF101. We saw a reduction in hallucination loss by a little more than 3%, which means that the hallucinated

representations were closer to the true spatiotemporal representations. Similarly, there was a slight classification improvement, but with a 67% increase in computation time.

The long-term action results are presented in Table 5b for MTL-AQA. Like with short-term actions, there was an improvement when using two frames. The percent of reduction in $\mathcal{L}_{hallu}$ was better than the short-term case, and dive classification was improved across all components (except **AS**, which was saturated).

Table 5. (**a**) Single-frame vs. two-frame on UCF-101. (**b**) Single-frame vs. two-frame: MTL-AQA dive classification. $\mathcal{L}_{hallu}$: lower is better.

(**a**)

Method	$\mathcal{L}_{hallu}$ ($\times 10^{-3}$)	**Accu**	**Time/Inf.**
HalluciNet(1f)	3.3	68.60	3.54 ms
HalluciNet(2f)	3.2 ($\downarrow$ 3.08%)	69.55	5.91 ms

(**b**)

Method	$\mathcal{L}_{hallu}$ ($\times 10^{-3}$)	**Accuracies**				
		P	**AS**	**RT**	**SS**	**TW**
HalluciNet(1f)	4.2	89.24	99.72	94.62	86.69	87.54
HalluciNet(2f)	3.9 ($\downarrow$ 8.05%)	**92.35**	99.72	**96.32**	**89.90**	**90.08**

Discussion:

Despite the lower mean hallucination error in the short-term case, the reduction rate was larger for the long-term actions. We believe this is due to the inherent difficulty of the classification task. In UCF-101, action classes are more semantically distinguishable, which makes it easier (e.g., archery vs. applying_makeup) to hallucinate and reason about the immediate future from a single image, while in the MTL-AQA dive classification case, the action evolution can be confusing or tricky to predict from a single image. An example is trying to determine the direction of rotation—it is difficult to determine if it is forward or backward with a snapshot devoid of motion. Moreover, differences between dives are more subtle. The tasks of counting somersaults and twists need accuracy up to half a rotation. As a result, short-term hallucination is more difficult—it is difficult to determine if it is a full or half rotation. While the the two-frame HalluciNet can extract some low-level motion cues to resolve ambiguity, the impact is tempered in UCF-101, which has less motion dependence. Consequently, there is comparatively more improvement in MTL-AQA, where motion (e.g., speed of rotation to distinguish between full/half rotation) is more meaningful to the classification task.

4.5. Utility of Hallucination Task in Pretraining

To determine if the hallucination task positively affects pretraining, we conducted an experiment on the downstream task of dive classification on the MTL-AQA dataset. In Experiment Section 4.2.1 the backbone network was trained on the UCF-101 action classification dataset; however, the hallucination task was not utilized during that pretraining. Table 6 summarizes the results of pretraining with and without the hallucination for dive classification. The use of hallucination during pretraining provided better initialization for both the vanilla 2D-CNN and HalluciNet, which led to improvements in almost every category besides rotation (RT) for HalluciNet. Additionally, HalluciNet training had the best performance for each dive class, indicating its utility both in pretraining network initialization and task-specific training.

Table 6. Utility of hallucination task in pretraining. Best performances are underlined.

Model	Pretraining w/ Hallucination	Accuracies				
		P	**AS**	**RT**	**SS**	**TW**
2D-CNN	No	90.08	99.43	92.07	83.00	86.69
	Yes	**92.35**	**99.72**	**94.33**	**86.97**	**88.95**
HalluciNet	No	89.52	99.43	**96.32**	86.12	88.10
	Yes	**92.92**	**99.72**	95.18	**88.39**	**91.22**

5. Conclusions

Although 3D-CNNs extract richer spatiotemporal features than the spatial features from 2D-CNNs, this comes at a considerably higher computational cost. We proposed a simple solution to approximate (hallucinate) spatiotemporal representations (computed by a 3D-CNN), using a computationally lightweight 2D-CNN with a single frame. Hallucinating spatiotemporal representations, instead of actually computing them, dramatically lowers the computational cost (only 6% of 3D-CNN time in our experiments), which makes deployment on edge devices feasible. In addition, by using only a single frame, rather than 16, the communication bandwidth requirements are lowered. Besides these practical benefits, we found that the hallucination task, when used in a multitask learning setting, provides a strong supervisory signal, which helps in (1) actions with short- and long-term dynamics, (2) dynamic scene recognition (non-action task), and (3) improving pretraining for downstream tasks. We showed the hallucination task across various base CNNs. Our hallucination task is a plug-and-play module, and we suggest future works to leverage the hallucination task for action as well as non-action tasks.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
2. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. The kinetics human action video dataset. *arXiv* **2017**, arXiv:1705.06950.
3. Hara, K.; Kataoka, H.; Satoh, Y. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6546–6555.
4. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. Slowfast networks for video recognition. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6202–6211.
5. Parmar, P.; Tran Morris, B. Learning to score olympic events. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 20–28.
6. Parmar, P.; Morris, B.T. What and How Well You Performed? A Multitask Learning Approach to Action Quality Assessment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Angeles, CA, USA, 16–20 June 2019; pp. 304–313.
7. Li, R.; Fu, H.; Zheng, Y.; Lo, W.L.; Jane, J.Y.; Sit, C.H.; Chi, Z.; Song, Z.; Wen, D. Automated fine motor evaluation for developmental coordination disorder. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *27*, 963–973. [CrossRef] [PubMed]

8. Sardari, F.; Paiement, A.; Hannuna, S.; Mirmehdi, M. VI-Net—View-Invariant Quality of Human Movement Assessment. *Sensors* **2020**, *20*, 5258. [CrossRef] [PubMed]
9. Lei, Q.; Zhang, H.B.; Du, J.X.; Hsiao, T.C.; Chen, C.C. Learning effective skeletal representations on RGB video for fine-grained human action quality assessment. *Electronics* **2020**, *9*, 568. [CrossRef]
10. Doughty, H.; Mayol-Cuevas, W.; Damen, D. The Pros and Cons: Rank-aware temporal attention for skill determination in long videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Angeles, CA, USA, 16–20 June 2019; pp. 7862–7871.
11. Ghanem, B.; Niebles, J.C.; Snoek, C.; Heilbron, F.C.; Alwassel, H.; Escorcia, V.; Krishna, R.; Buch, S.; Dao, C.D. The activitynet large-scale activity recognition challenge 2018 summary. *arXiv* **2018**, arXiv:1808.03766.
12. Zhou, Y.; Sun, X.; Zha, Z.J.; Zeng, W. Mict: Mixed 3d/2d convolutional tube for human action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 449–458.
13. Wang, H.; Lin, J.; Wang, Z. Design Light-weight 3D Convolutional Networks for Video Recognition Temporal Residual, Fully Separable Block, and Fast Algorithm. *arXiv* **2019**, arXiv:1905.13388.
14. Zhang, H.; Li, Y.; Wang, P.; Liu, Y.; Shen, C. RGB-D Based Action Recognition with Light-weight 3D Convolutional Networks. *arXiv* **2018**, arXiv:1811.09908.
15. Hadidi, R.; Cao, J.; Xie, Y.; Asgari, B.; Krishna, T.; Kim, H. Characterizing the Deployment of Deep Neural Networks on Commercial Edge Devices. In Proceedings of the IEEE International Symposium on Workload Characterization, Orlando, FL, USA, 3–5 November 2019.
16. Walker, J.; Gupta, A.; Hebert, M. Patch to the future: Unsupervised visual prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 3302–3309.
17. Walker, J.; Gupta, A.; Hebert, M. Dense optical flow prediction from a static image. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2443–2451.
18. Pintea, S.L.; van Gemert, J.C.; Smeulders, A.W. Déja vu. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 172–187.
19. Gao, R.; Xiong, B.; Grauman, K. Im2flow: Motion hallucination from static images for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5937–5947.
20. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Li, F.-F. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 1725–1732.
21. He, Y.; Shirakabe, S.; Satoh, Y.; Kataoka, H. Human action recognition without human. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 11–17.
22. Yuen, J.; Torralba, A. A data-driven approach for event prediction. In *Proceedings of the European Conference on Computer Vision, 2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 707–720.
23. Kitani, K.M.; Ziebart, B.D.; Bagnell, J.A.; Hebert, M. Activity forecasting. In *Proceedings of the European Conference on Computer Vision, 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 201–214.
24. Finn, C.; Goodfellow, I.; Levine, S. Unsupervised learning for physical interaction through video prediction. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 64–72.
25. Walker, J.; Doersch, C.; Gupta, A.; Hebert, M. An uncertain future: Forecasting from static images using variational autoencoders. In *Proceedings of the European Conference on Computer Vision, 2016*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 835–851.
26. Koppula, H.S.; Saxena, A. Anticipating human activities using object affordances for reactive robotic response. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 14–29. [CrossRef] [PubMed]
27. Vondrick, C.; Pirsiavash, H.; Torralba, A. Anticipating visual representations from unlabeled video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 98–106.
28. Vondrick, C.; Pirsiavash, H.; Torralba, A. Generating videos with scene dynamics. *Adv. Neural Inf. Process. Syst.* **2016**,29, 613–621.
29. Bilen, H.; Fernando, B.; Gavves, E.; Vedaldi, A.; Gould, S. Dynamic image networks for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3034–3042.
30. Vondrick, C.; Torralba, A. Generating the future with adversarial transformers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1020–1028.
31. Wang, X.; Farhadi, A.; Gupta, A. Actions˜ transformations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2658–2667.
32. Hoffman, J.; Gupta, S.; Darrell, T. Learning with side information through modality hallucination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 826–834.
33. Zhang, Z.; Kuang, Z.; Luo, P.; Feng, L.; Zhang, W. Temporal Sequence Distillation: Towards Few-Frame Action Recognition in Videos. In Proceedings of the 2018 ACM Multimedia Conference on Multimedia Conference, Seoul, Korea, 22–26 October 2018; pp. 257–264.
34. Bhardwaj, S.; Srinivasan, M.; Khapra, M.M. Efficient Video Classification Using Fewer Frames. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Angeles, CA, USA, 16–20 June 2019; pp. 354–363.

35. Crasto, N.; Weinzaepfel, P.; Alahari, K.; Schmid, C. MARS: Motion-Augmented RGB Stream for Action Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
36. Stroud, J.C.; Ross, D.A.; Sun, C.; Deng, J.; Sukthankar, R. D3D: Distilled 3D Networks for Video Action Recognition. *arXiv* **2018**, arXiv:1812.08249.
37. Sun, L.; Jia, K.; Yeung, D.Y.; Shi, B.E. Human action recognition using factorized spatio-temporal convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4597–4605.
38. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lake City, UT, USA, 18–22 June 2018; pp. 6450–6459.
39. Qiu, Z.; Yao, T.; Mei, T. Learning spatio-temporal representation with pseudo-3d residual networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5533–5541.
40. Xie, S.; Sun, C.; Huang, J.; Tu, Z.; Murphy, K. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 305–321.
41. Lee, M.; Lee, S.; Son, S.; Park, G.; Kwak, N. Motion feature network: Fixed motion filter for action recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 387–403.
42. Lin, J.; Gan, C.; Han, S. Tsm: Temporal shift module for efficient video understanding. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 7083–7093.
43. Zhou, B.; Andonian, A.; Oliva, A.; Torralba, A. Temporal relational reasoning in videos. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 803–818.
44. Ji, J.; Buch, S.; Soto, A.; Carlos Niebles, J. End-to-end joint semantic segmentation of actors and actions in video. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 702–717.
45. Gavrilyuk, K.; Ghodrati, A.; Li, Z.; Snoek, C.G. Actor and action video segmentation from a sentence. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lake City, UT, USA, 18–22 June 2018; pp. 5958–5966.
46. Yan, Y.; Xu, C.; Cai, D.; Corso, J.J. Weakly supervised actor-action segmentation via robust multi-task ranking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1298–1307.
47. Kalogeiton, V.; Weinzaepfel, P.; Ferrari, V.; Schmid, C. Joint learning of object and action detectors. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4163–4172.
48. Xu, C.; Corso, J.J. Actor-action semantic segmentation with grouping process models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3083–3092.
49. Xu, C.; Hsieh, S.H.; Xiong, C.; Corso, J.J. Can humans fly? action understanding with multiple classes of actors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2264–2273.
50. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
51. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
52. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in Pytorch. 2017. Available online: https://openreview.net/forum?id=BJJsrmfCZ (accessed on 16 August 2021).
53. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
54. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.
55. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
56. Nibali, A.; He, Z.; Morgan, S.; Greenwood, D. Extraction and Classification of Diving Clips from Continuous Video Footage. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 94–104.
57. Pirsiavash, H.; Vondrick, C.; Torralba, A. Assessing the quality of actions. In *Proceedings of the European Conference on Computer Vision, 2014*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 556–571.
58. Feichtenhofer, C.; Pinz, A.; Wildes, R.P. Temporal residual networks for dynamic scene recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4728–4737.
59. Theriault, C.; Thome, N.; Cord, M. Dynamic scene classification: Learning motion descriptors with slow features analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2603–2610.
60. Feichtenhofer, C.; Pinz, A.; Wildes, R.P. Bags of spacetime energies for dynamic scene recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 2681–2688.
61. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *arXiv* **2014**, arXiv:1406.2199.