Deep Analysis of CNN-based Spatio-temporal Representations for Action Recognition

Chun-Fu (Richard) Chen[1,†], Rameswar Panda[1,†], Kandan Ramakrishnan[1],
Rogerio Feris[1], John Cohn[1], Aude Oliva[2], Quanfu Fan[1,†]
†: Equal Contribution
[1]MIT-IBM Watson AI Lab, [2]Massachusetts Institute of Technology

Abstract

In recent years, a number of approaches based on 2D or 3D convolutional neural networks (CNN) have emerged for video action recognition, achieving state-of-the-art results on several large-scale benchmark datasets. In this paper, we carry out in-depth comparative analysis to better understand the differences between these approaches and the progress made by them. To this end, we develop an unified framework for both 2D-CNN and 3D-CNN action models, which enables us to remove bells and whistles and provides a common ground for fair comparison. We then conduct an effort towards a large-scale analysis involving over 300 action recognition models. Our comprehensive analysis reveals that a) a significant leap is made in efficiency for action recognition, but not in accuracy; b) 2D-CNN and 3D-CNN models behave similarly in terms of spatio-temporal representation abilities and transferability. Our codes are available at `https://github.com/IBM/action-recognition-pytorch`.

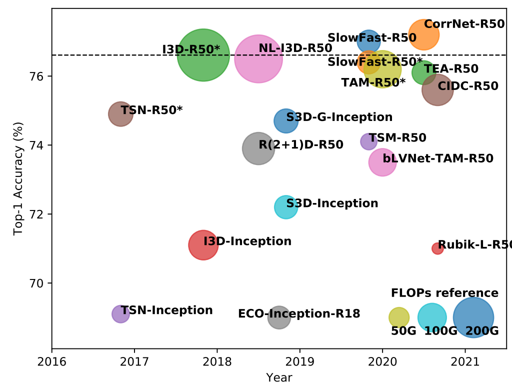

Figure 1: Recent progress of action recognition on Kinetics-400 (only models based on InceptionV1 and ResNet50 are included). Models marked with $*$ are re-trained and evaluated (see Section 6.2) while others are from the existing literature. The size of a circle indicates the 1-clip FLOPs of a model. With temporal pooling turned off, I3D performs on par with the state-of-the-art approaches. Best viewed in color.

1. Introduction

With the recent advances in convolutional neural networks (CNNs) [59, 24] and the availability of large-scale video benchmark datasets [31, 44], deep learning approaches have dominated the field of video action recognition by using 2D-CNNs [68, 38, 8] or 3D-CNNs [2, 22, 10] or both [40, 57]. The 2D CNNs perform temporal modeling independent of 2D spatial convolutions while their 3D counterparts learn space and time information jointly by 3D convolution. These methods have achieved state-of-the-art performance on multiple large-scale benchmarks such as Kinetics [31] and Something-Something [20].

Although CNN-based approaches have made impressive progress in action recognition, there are several fundamental questions that still largely remain unanswered in the field. For example, what contributes to improved spatio-temporal representations of these recent approaches? Do these approaches enable more effective temporal modeling, the crux of the matter for action recognition? Furthermore, there seems no clear winner between 2D-CNN and 3D-CNN approaches in terms of accuracy. 3D models report better performance than 2D models on Kinetics while the latter are superior on Something-Something. How differently do these two types of models behave with regard to spatial-temporal modeling of video data?

We argue that the difficulty of understanding the recent progress on action recognition is mainly due to the lack of fairness in performance evaluation related to datasets, backbones and experimental practices. In contrast to image recognition where ImageNet [4] has served as a gold-standard benchmark for evaluation, there are at least 4∼5 popular action datasets widely used for evaluation (see Figure 2). While Kinetics-400 [31] has recently emerged as a primary benchmark for action recognition, it is known to be strongly biased towards spatial modeling, thus being inappropriate for validating a model's capability of spatio-temporal modeling. In addition, there seems to be a tendency in current research to overly focus on pursuing state-of-the-art (SOTA) performance, but overlooking other important factors such as the backbone networks and the number of input frames. For instance, I3D [2] based on 3D-InceptionV1 has become a *"gatekeeper"* baseline to compare with for any recently proposed approaches of action recognition. However such comparisons are often unfair against stronger backbones such as ResNet50 [24]. As shown in Figure 1, I3D, with ResNet50 as backbone, performs comparably with or outperforms many recent methods that are claimed to be better. As a result, such evaluation are barely informative w.r.t whether the improved results of an approach come from a better backbone or the algorithm itself. As discussed in Section 3, performance evaluation in action recognition may be further confounded by many other issues such as variations in training and evaluation protocols, model inputs and pretrained models.

In light of the great need for better understanding of CNN-based action recognition models, in this paper we provide a common ground for comparative analysis of 2D-CNN and 3D-CNN models without any bells and whistles. We conduct comprehensive experiments and analysis to compare several representative 2D-CNN and 3D-CNN methods on three large-scale benchmark datasets. Our main goal is to deliver deep understanding of the important questions brought up above, especially, *a) the current progress of action recognition and b) the differences between 2D-CNN and 3D-CNN methods w.r.t spatial-temporal representations of video data.* Our systematic analysis provides insights to researchers to understand spatio-temporal effects of different action models across backbone and architecture and will broadly simulate discussions in the community regarding a very important but largely neglected issue of fair comparison in video action recognition.

The main contributions of our work as follows:

- **A unified framework for Action Recognition.** We present a unified framework for 2D-CNN and 3D-CNN approaches and implement several representative methods for comparative analysis on three standard action recognition benchmark datasets.
- **Spatio-Temporal Analysis.** We systematically compare 2D-CNN and 3D-CNN models to better understand the differences and spatio-temporal behavior of these models. Our analysis leads to some interesting findings as follows: a) Temporal pooling tends to suppress the efficacy of temporal modeling in an action model, but surprisingly provides a significant performance boost to TSN [68]; b) By removing non-structural differences between 2D-CNN and 3D-CNN models, they behave similarly in terms of spatio-temporal representation abilities and transferability.
- **Benchmarking of SOTA Approaches.** We thoroughly benchmarked several SOTA approaches and compared them with I3D. Our analysis reveals that I3D still stays on par with SOTA approaches in terms of accuracy (Figure 1) and the recent advance in action recognition is mostly on the efficiency side, not on accuracy. Our analysis also suggests that the input sampling strategy taken by a model (i.e. uniform or dense sampling) should be considered for fairness when comparing two models (Section 6.2).

2. Related Work

Video understanding has made rapid progress with the introduction of a number of large-scale video datasets such as Kinetics [31], Sports1M [30], Moments-In-Time [44], and YouTube-8M [1]. A number of models introduced recently have emphasized the need to efficiently model spatio-temporal information for video action recognition.

Most successful deep architectures for action recognition are usually based on two-stream model [54], processing RGB frames and optical-flow in two separate CNNs with a late fusion in upper layers [30]. Two-stream approaches have been used in different action recognition methods [3, 6, 19, 75, 56, 63, 70, 66, 11, 12]. Another straightforward but popular approach is the use of 2D-CNN to extract frame-level features and then model the temporal causality. For example, TSN [68] propose consensus module to aggregate features; on the other hand, TRN [77] use bag of features to model relationship between frames. While TSM [38] shifts part of the channels along temporal dimension, thereby allowing for information to be exchanged among neighboring frames, TAM [8] is based on depthwise 1×1 convolutions to capture temporal dependencies across frames effectively. Different methods for temporal aggregation of feature descriptors have also been proposed [13, 35, 73, 66, 48, 16, 15]. More complex approaches have also been investigated for capturing long-range dependencies, e.g. non-local neural networks [69].

Another approach is to use 3D-CNN, which extends the success of 2D models in image recognition [28] to recognize actions in videos. For example, C3D [60] learns 3D ConvNets which outperforms 2D CNNs through the use of large-scale video datasets. Many variants of 3D-CNNs are introduced for learning spatio-temporal features such as I3D [2] and ResNet3D [22]. 3D CNN features were also demonstrated to generalize well to other vision tasks, such as action detection [52], video captioning [45], action localization [47], and video summarization [46]. Nonetheless, as 3D convolution leads high computational load, few works aim to reduce the complexity by decomposing the 3D convolution into 2D spatial convolution and 1D temporal convolution, e.g., P3D [50], S3D [72], R(2+1)D [62], or incorporating group convolution [61]; or using a combination of 2D-CNN and 3D-CNN [79]. Furthermore, SlowFast network employs two pathways to capture short-term and long-term temporal information [10] by processing a video at both slow and fast frame rates. Beyond that, Timeception applies the Inception concept in the temporal domain for capturing long-range temporal dependencies [26]. Feichtenhofer [9] finds efficient networks by extending 2D architectures through a stepwise expansion approach over the key variables such as temporal duration, frame rate, spatial resolution, network width, etc. Leveraging weak supervision [14, 67, 33] or distillation [18] is another recent trend in action recognition. Few works have assessed the importance of temporal information in a video, e.g., Sigurdsson *et.al* analyzed performance per action category based on different levels of object complexity, verb complexity, and motion [53]. They state that to differentiate temporally similar but semantically different videos, its important for models to develop temporal understanding. Huang *et. al* analyzed the effect of motion via an ablation analysis on C3D model [25]. Nonetheless, these works only study a limited set of backbone and temporal modeling methods.

3. Challenges of Evaluating Action Models

The first challenge in evaluating action models stem from the fact that unlike ImageNet for image classification, action recognition does not have one dataset widely used for every paper. As shown in Figure 2, the most popular Kinetics-400 is used by around 60% papers[1]. On the other hand, Something-Something (V1 and V2), which has very different temporal characteristic from Kinetics-400, is also used by about 50% papers. Furthermore, two successors of the Kinetics-400 datasets, Kinetics-600 and Kinetics-700 are released recently. It is difficult to evaluate different methods if they do not test on common datasets. We further check those 37 papers how do they compare the performance in their paper [60, 68, 2, 17, 50, 78, 58, 77, 5, 76, 22, 65, 34, 79, 72, 62, 69, 23, 42, 29, 49, 38, 8, 36, 26, 41, 10, 61, 71, 39, 57, 51, 74, 7, 64, 9, 37]. We evaluate those papers from four aspects, including backbone, input length, training protocol and evaluation protocol. Figure 2 shows the summary of how papers compare to others differently.

Backbone. From our analysis, we observe that about 70% papers compare results with different backbones (e.g., most of the papers use ResNet50 as backbone but compare with I3D [2] which uses InceptionV1 as the backbone). Comparing action models with different types of backbones can often lead to incorrect conclusions, also making harder to evaluate the advantage of the proposed temporal modeling. For example, using stronger backbone for I3D, it improves the results by 4.0% on Kinetics-400 (see Figure 7).

[1] Kinetics-400 dataset is available after 2017, the used rate increases to 69% if only the papers published after 2017 are counted.

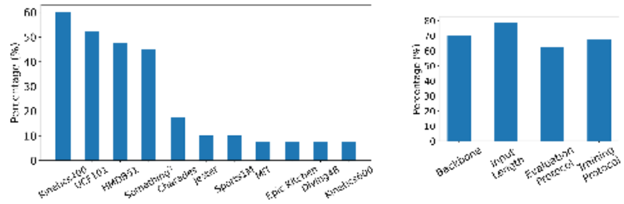

Figure 2: Statistics collected from 37 action recognition papers from 2015 to 2020. Left: Used datasets. Right: Ratio of papers used different settings to compare with others.

Input Length. Figure 2 shows that about 80% of the papers use different number of frames for comparison. It is because each method could prefer different frame numbers; however, comparing under different number of frames could favor either the proposed method or the reference method.

Training Protocol. Due to recent advances in technology, It is often easier to train action recognition models for a very long time (epochs) which was not possible a few years ago, indicating that old methods might not be well-trained. Furthermore, many works reuse the ImageNet weights to initialize the models while others are not. It raises the concern that does the gain comes from different training protocol. Based on our analysis, about 60% of the papers use different protocols to train action recognition models.

Evaluation Protocol. As the models are trained under different sampling strategies and input lengths, a model is used to take more than one clip from a video for prediction. Hence, different evaluation protocol could lead unclear comparison. About 60% papers evaluated models differently when comparing to others.

4. 2D-CNN and 3D-CNN Approaches

To address the above mentioned issue for fair comparison, we analyze several popular 2D-CNN and 3D-CNN approaches for action recognition, including I3D [2], ResNet3D [21], S3D [72], R(2+1)D [62], TSN [68] and TAM [8]. These approaches not only yield competitive results on popular large-scale datasets, but also widely serve as fundamental building blocks for many other successive approaches such as SlowFast [10] and CSN [61].

Among these approaches, I3D and ResNet3D are pure 3D-CNN models, differing only in backbones. S3D and R(2+1)D factorize a 3D convolutional filter into a 2D spatial filter followed by a 1D temporal filter. In such a sense, they are architecturally similar to 2D models. However, we categorize them into 3D-CNN models as their implementations are based on 3D convolutions. While TSN rely only on 2D convolution without temporal modeling, TAM, another 2D-CNN approach, adds efficient depthwise temporal aggregation on top of TSN, which shows strong results on Something-Something [8]. Finally, since SlowFast is arguably one of the best approaches on Kinetics, we use it

Approach	Model Input	Input Sampling	Backbone	Temporal Pooling	Spatial Module	Temporal Aggregation	Initial Weights
I3D [2] R3D [21]	4D	Dense	InceptionV1 ResNet	Y	3D Conv.	3D Conv.	Inflation
S3D [72] R(2+1)D [62]			InceptionV1 ResNet		2D Conv.	1D Conv.	Inflation Scratch
TAM [8] TSN [68]	3D	Uniform	bLResNet InceptionV1	N	2D Conv.	1D dw Conv. None	ImageNet ImageNet

Table 1: 2D-CNN and 3D-CNN approaches in our study.

as a reference to SOTA results. Apart from using different types of convolutional kernels, 2D and 3D models differ in a number of other aspects, including model input, temporal pooling, and temporal aggregation, as shown in Table 1.

The differences between 2D-CNN and 3D-CNN approaches make it a challenge to compare these approaches. To remove the bells and whistles and ensure a fair comparison, we show in Figure 3 that 2D and 3D models can be represented by a general framework. Under such a framework, an action recognition model is viewed as a sequence of stacked spatio-temporal modules with temporal pooling optionally applied. Thus what differentiates a model from another boils down to only its spatio-temporal module. We re-implemented all the approaches used in our comparison under this framework, which allows us to test an approach flexibly using different configurations such as backbone, temporal pooling and temporal aggregation. For example, in S3D-ResNet (i.e., R(2+1)D), we do not expand the channel dimension between spatial and temporal convolution to keep it align to S3D [72]. More details on the models and implementations can be found in the Supplemental.

5. Datasets, Training, Evaluation Protocols

To ensure fair comparison and facilitate reproduciblity, we train all the models using the same data preprocessing, training protocol, and evaluation protocol. Below we provide a brief description and refer the reader to the Supplemental for more details including the source codes.

Datasets. We choose Something-Something V2 (*SSV2*), Kinetics-400 (*Kinetics*) and Moments-in-time (*MiT*) for our experiments. We also create a mini version of each dataset: *Mini-SSV2* and *Mini-Kinetics* account for half of their full datasets by randomly selecting half of the categories of *SSV2* and *Kinetics*. *Mini-MiT* is provided on the official *MiT* website, consisting of 1/8 of videos in the full dataset.

Training. Following [8], we progressively train the models using different input frames. Let $K_i \in [8, 16, 32, 64]$ where $i = 1 \dots 4$. We first train a starter model using 8 frames. The model is either inflated with (e.g., I3D) or initialized from (e.g., TAM) its corresponding ImageNet pre-trained model. We then finetune the model using more frames K_i from the model using K_{i-1} frames.

Evaluation. There are two major evaluation metrics for video action recognition: clip-level accuracy and video-level accuracy. Clip-level accuracy is prediction by feeding

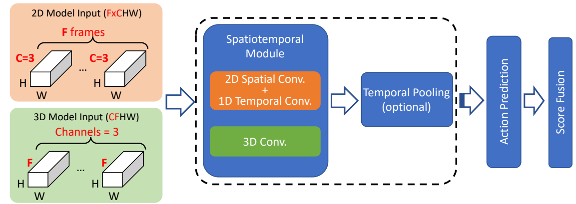

Figure 3: A general framework for 2D-CNN and 3D-CNN approaches of video action recognition. A video action recognition model can be viewed as a sequence of stacked spatio-temporal modules. The input frames are formed as a 3D tensor for 2D models and 4D tensor for 3D models.

a single clip into the network and video-level accuracy is the combined predictions of *multiple* clips; thus, the video-level accuracy is usually higher than the clip-level accuracy. By default, we report the clip-level accuracy.

6. Experimental Results and Analysis

In this section, we provide a detailed analysis on the performance of 2D and 3D models (Section 6.1), SOTA results and transferability (Section 6.2) and their spatio-temporal effects (Section 6.3). For clarity, from now onwards, we refer to each of *I3D*, *S3D* and *TAM* as one type of spatio-temporal module illustrated in Figure 3. We name a specific model by *module-backbone[-tp]* where *tp* indicates that temporal pooling is applied. For example, I3D-ResNet18-tp is a 3D model based on ResNet18 with temporal pooling. To verify the correctness of our implementation, we trained a I3D-InceptionV1 as the original paper [2], and find that our model achieves 73.1% top-1 accuracy, which is 2% better than the result reported in the original paper. It clearly justifies the results conducted by our setup is reliable.

6.1. Performance Analysis on Mini Datasets

For each spatio-temporal module, we experiment with 3 backbones (InceptionV1, ResNet18 and ResNet50) and two scenarios (w/ and w/o temporal pooling) on three datasets. In each case, 8, 16, 32 and 64 frames are considered as input. This results in a total of $4 \times 3 \times 2 \times 3 \times 4 = 288$ models to train, many of which haven't been explored in the original papers. Since temporal pooling is detrimental to model performance (see Figure 6), our analysis in this work mainly focus on models w/o temporal pooling unless otherwise specified. Figure 4 reports the clip-level top-1 accuracies w/o temporal pooling for all models. We refer readers to the Supplemental for the results w/ temporal pooling.

Backbone Network and Input Length. As seen from Figure 4, regardless of the spatiotemporal modules used, there is a general tendency that ResNet50 $>$ InceptionV1 $>$ ResNet18 w.r.t their overall spatiotemporal representation capability. Longer input frames tend to produce better results; however, the performance improvement does not

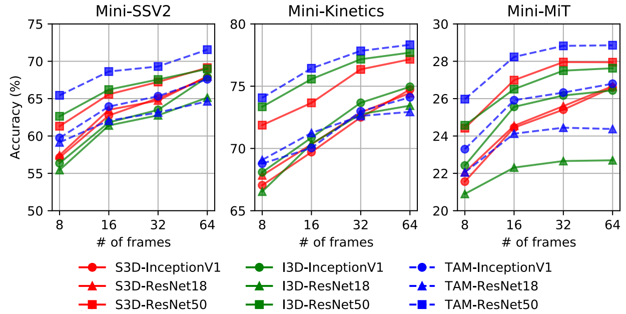

Figure 4: Top-1 accuracy of all the compared models without temporal pooling on three mini-datasets. The video architectures are separated by color while the backbones by symbol.

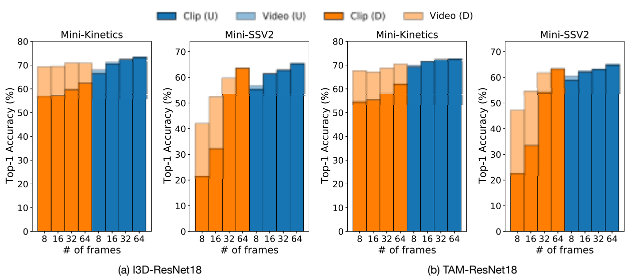

Figure 5: Performance comparison between *Uniform Sampling (U)* and *Dense Sampling (D)*. (a) I3D-ResNet18 (b) TAM-ResNet18. Both models do not include temporal pooling. Solid bars are the clip-level accuracy while transparent bars indicates the improvement by the video-level (multi-clip) evaluation.

seem significant after 32 frames on all three datasets.

Input Sampling. Two sampling strategies are widely adopted in action recognition to create model inputs. The first one, *Uniform sampling*, which is often seen in 2D models, divides a video into multiple equal-length segments and then randomly selects one frame from each segment. The other method used by 3D models, *dense sampling*, instead directly takes a set of continuous frames as the input. It is not clear, though, why these two types of models prefer different inputs. Figure 5 shows that uniform sampling (blue) yields better clip-level accuracies than dense sampling (orange) under all circumstances. This is not surprising as dense sampling only uses part of the test video in the clip-level evaluation. Even though the video-level evaluation boosts the performance of dense sampling by 6%∼15% on *Mini-Kinetics* and 5%∼20% on *Mini-SSV2*, its computational needs are increased proportionally, e.g., 10 clips used in Figure 5 to get video-level accuracy, increases the FLOPs by ten folds. Such costs make it inappropriate in practice. Thus, all our analysis is based on uniform sampling and clip-level evaluation unless otherwise stated. We will further analyze the effect of input sampling strategies in Section 6.2 based on the results from full datasets.

Temporal Pooling. Temporal pooling is usually applied to 3D models to reduce computational complexity. It is known that temporal pooling negatively affects model performance. Such effects, however, have not been well understood in the literature. Figure 6 shows the performance gaps between models w/ and w/o temporal pooling across different backbones and architectures. As can be seen, temporal pooling in general counters the effectiveness of temporal modeling and hurts the performance of action models, just like what spatial pooling does to object recognition and detection. For this reason, more recent 3D-CNN approaches such as SlowFast [10] and X3D [9] drop temporal pooing and rely on other techniques for reducing computation. Similarly, one important reason for the prior finding in [27] that 3D models are inferior to C2D (pure spatial models) on *Kinetics* and *MiT* is because their comparisons neglect the negative impact of temporal pooling on 3D models. As shown in Section 6.2, I3D w/o temporal pooling is competitively comparable with the SOTA approaches.

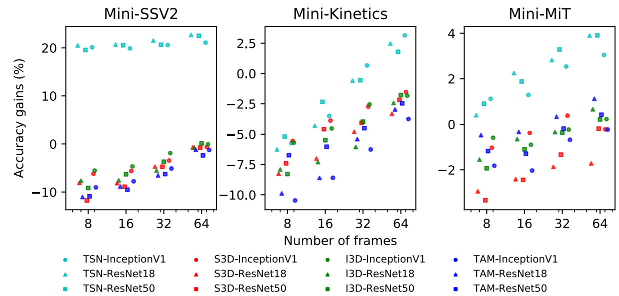

Figure 6: Accuracy gain of the models with temporal pooling w.r.t. the models without temporal pooling. Temporal pooling significantly hurts the performance of all models except TSNs.

Interestingly, TSN is the only architecture benefiting from temporal pooling, demonstrating a large boost in performance on *Mini-SSV2* (>20%) and *Mini-MiT* (3%∼5%). Also, as the number of input frames increases, the improvement is more pronounced. On *Mini-Kinetics*, even though TSN is also negatively affected by temporal pooling , it suffers the least and starts seeing positive gains after 32 frames. To further confirm that, we trained a 32-frame TSN model with temporal pooling on Kinetics. This model (TSN-R50$*$ in Figure 1) achieves a top-1 accuracy of 74.9%, 5.1% higher than the version w/o temporal pooling and only about 2.0% shy from the SOTA results. We interpret *temporal pooling as a simple form of exchanging information across frames*, which empowers TSN with the ability of temporal modeling. The consistent improvements by temporal pooling across all the datasets provide strong evidence that temporal modeling is necessary for video action recognition, even for datasets like Kinetics where temporal information has been shown less crucial for recognition.

6.2. Benchmarking of SOTA Approaches

Results on Full Datasets. I3D based on InceptionV1 has been used as an important baseline by many papers to show-

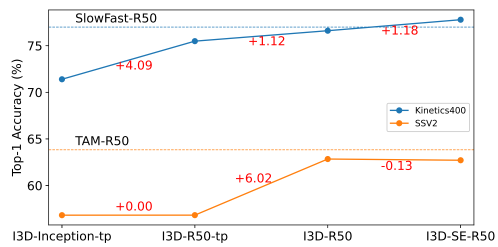

Figure 7: Performance of I3D models by changing the backbone (I3D-R50-tp), removing temporal pooling (I3D-R50-tp) and adding squeeze-excitation modules (I3D-SE-R50) on *Kinetics* and *SSV2*. Red numbers indicate performance changes. All models are trained with 32 frames and evaluated using 3 × 10 clips on *Kinetics*, and 3 × 2 clips on *SSV2*, respectively.

case their progress. However, the results of I3D on the mini datasets, especially the unexpectedly significant impact of temporal pooling, seem to suggest that the spatio-temporal modeling capability of I3D has been underestimated by the field. To more precisely understand the recent progress in action recognition, we further conduct a more rigorous benchmarking effort including I3D, TAM and SlowFast on the full datasets. I3D was the prior SOTA method while SlowFast [10] and TAM [8], both of which have official codes released, are competitively comparable with existing SOTA methods. To ensure apple-to-apple comparison, we follow the same training settings of SlowFast to train all the models using 32 frames as input. During evaluation, we use 3 × 10 clips for *Kinetics* and *MiT*, and 3 × 2 clips for *SSV2*.

We first augment original I3D by a stronger backbone ResNet50 and turning off temporal pooling. As shown in Figure 7, ResNet50 alone pushes up the accuracy of I3D by 4.0% on *Kinetics*, and removing temporal pooling adds another 1.1% performance gain, putting I3D on par with SlowFast in terms of top-1 accuracy. Further inserting Squeeze-Excitation modules into I3D makes it surpass SlowFast by 0.8%. On *SSV2*, a stronger backbone provides I3D little benefit in accuracy, but removing temporal pooling boosts the performance substantially by 6%, making I3D comparable to TAM. Table 2 provides more detailed results in this experiment. In summary, I3D-ResNet50 demonstrates impressive results, staying on par with state-of-the-art approaches in accuracy on all three datasets. The fact that I3D remains very strong across multiple large-scale datasets suggests that the recent progress of action recognition in terms of accuracy is largely attributed to the use of more powerful backbone networks, but not the improved spatio-temporal modeling as expected. Nevertheless, we do observe that recent approaches such as X3D [9] have made a large leap ahead in efficiency (FLOPs) compared to I3D. Moreover, SlowFast performs worse than I3D and TAM on *SSV2* on the Something-Something dataset. We speculate that this could be related to: (I) that the slow pathway only uses temporal convolutions after stage4 of ResNet, which weakens its temporal modeling capability; and (II) that the two-stream architecture is less effective in capturing temporal dependencies in such a highly temporal dataset.

Model	Pretrain dataset	FLOPs	Dataset *Kinetics*	*SSV2*	*MiT*
I3D-ResNet50	ImageNet	335.3G	76.61	62.84	31.21
I3D-ResNet50	None	335.3G	76.54	–	–
TAM-ResNet50	ImageNet	171.5G	76.18	63.83	30.80
SlowFast-ResNet50-8×8† [10]	None*	65.7G	76.40	60.10	31.20
I3D-ResNet101	ImageNet	654.7G	77.80	64.29	–
TAM-ResNet101	ImageNet	327.1G	77.61	65.32	–
SlowFast-ResNet50-8×8‡ [10]	None*	65.7G	77.00	–	–
SlowFast-ResNet50-16×8‡ [10]	*Kinetics*	124.5G	–	63.0	–
CorrNet-ResNet50‡ [64]	None*	115G	77.20	–	–
SlowFast-ResNet101-8×8† [10]	None	125.9G	76.72	–	–
SlowFast-ResNet101-8×8‡ [10]	None	125.9G	78.00	–	–
SlowFast-ResNet101-16×8‡ [10]	None	213G	78.90	–	–
CSN-ResNet101‡ [61]	None*	83G	76.70	–	–
CorrNet-ResNet101‡ [64]	None*	224G	79.20	–	–
X3D-L‡ [9]	None*	24.8G	77.50	–	–
X3D-XL‡ [9]	None*	48.4G	79.10	–	–
AssembleNet-50^1 [51]	–	–	–	–	31.41
GST-ResNet101 [36]	ImageNet	–	–	–	32.40

*: Those networks cannot be initialized from ImageNet due to its structure.
†: Retrained by ourselves. ‡: reported by the authors of the paper. ^{1}: Use RGB + Flow.

Table 2: Performance of SOTA models.

Model	Pretrain	U-Sampling	D-Sampling
I3D-ResNet50	ImageNet	76.07	76.61
TAM-ResNet50	ImageNet	76.45	76.18
SlowFast-ResNet50-8×8	–	71.85	76.40

Table 3: Model performance on *Kinetics* based on uniform and dense sampling. Uniform sampling trained models are evaluated under 3 256×256 spatial crops and 2 clips.

Uniform Sampling vs Dense Sampling. We revisit the effect of input sampling on model performance and retrain all three approaches using uniform sampling on *Kinetics*. As shown in Table 3, the small difference between uniform and dense sampling results indicates that both I3D and TAM are flexible w.r.t model input. In contrast, uniform sampling is not as friendly as dense sampling to SlowFast, producing an accuracy ∼5% lower than dense sampling. We conjecture that this has to do with dual-path architecture of SlowFast. Such an architecture is primarily designed for efficiency and possibly less effective in learning spatial-temporal representations from sparsely sampled frames (i.e. 8-frame uniform sampling in this case). This also explains why SlowFast, when trained with uniform sampling, under performs by 2% ∼ 3% on *SSV2* in Table 2 in contrast to I3D and TAM.

Furthermore, Figure 8 (Left) shows model accuracy v.s. number of clips used for evaluation in uniform and dense sampling, respectively. As can be observed, the model performance with dense sampling is saturated quickly after 4-5 clips for both I3D and SlowFast. This suggests that the common practice in the literature of using 10 clips for dense sampling is often not necessary. As opposed to dense sampling, uniform sampling benefits slightly (i.e., for SlowFast) or little from multiple clips. This raises another pitfall

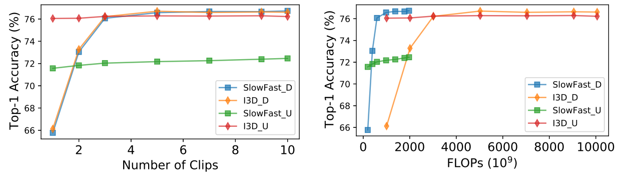

Figure 8: Model performance tested using 3 256×256 spatial crops and different number of clips. 'U': uniform sampling; 'D': dense sampling. Best viewed in color.

Model	Target dataset UCF101	HMDB51	Jester	*Mini-SSV2*
I3D-ResNet50	97.12	72.32	96.39	65.86
TAM-ResNet50	95.05	71.67	96.35	66.91
SlowFast-ResNet50-8×8	95.67	74.61	96.75	63.93

Table 4: Top-1 Acc. of Transferability study from *Kinetics*.

that is largely overlooked by the community when assessing model efficiency, i.e., the impact of input sampling. As shown in Figure 8 (Right), when putting I3D and SlowFast in a plot of accuracy v.s. FLOPs for comparison, the advantage of SlowFast over I3D is better and more fairly represented, i.e., when considering uniform sampling for I3D, SlowFast is only slightly more accurate but at the same efficiency in FLOPs. This clearly suggests that input sampling strategy of a model (i.e. uniform or dense) should factor in evaluation for fairness when comparing it to another model.

Model Transferability. We further compare the transferability of the three models trained above on four small-scale datasets including UCF101 [55], HMDB51 [32], Jester [43], and *Mini-SSV2*. We follow the same training setting in Section 5 and finetune 45 epochs with cosine annealing learning rate schedule starting with 0.01; furthermore, since those are 32-frame models, we trained the models with a batch size of 48 with synchronized batch normalization. Table 4 shows the results, indicating that all the three models have very similar performance (difference of less than 2%) on the downstream tasks. In particular, I3D performs on par with the SOTA approaches like TAM and SlowFast in transfer learning (e.g., I3D obtains the best accuracy of 97.12% on UCF101), which once again corroborates the fact that the improved spatio-temporal modeling is largely due the use of stronger backbones.

6.3. Analysis of Spatio-temporal Effects

It's generally believed that temporal modeling is the core for action recognition and state-of-the-art approaches can capture better temporal information. However, it has also been demonstrated on datasets such as *Kinetics* and *Moments-in-Time* (*MiT*) [44] that approaches purely based on spatial modeling [68, 44] can achieve very competitive results compared to more sophisticated spatio-temporal models. More recently, a paper [27] also shows that 2D models outperform their 3D counterparts on the *MiT* benchmark. These findings seem to imply that more complex temporal modeling is not necessary for "static" datasets such as *Kinetics* and *MiT*. We believe that lack of fairness in performance evaluation leads to confusion on understanding significance of temporal modeling for action recognition.

Dataset	Frames	InceptionV1 None	InceptionV1 I3D	InceptionV1 Conv.	InceptionV1 TAM	ResNet50 None	ResNet50 I3D	ResNet50 Conv.	ResNet50 TAM	ResNet50 TSM	ResNet50 NLN
Mini-SSV2	f=8	33.1	56.4	58.2	**59.7**	33.9	62.6	61.6	**65.4**	64.1	53.0
	f=16	34.7	61.8	63.7	**63.9**	35.3	66.2	65.7	**68.6**	67.4	55.0
Mini-Kinetics	f=8	**70.4**	68.1	68.3	68.8	72.1	73.3	71.5	**74.1**	**74.1**	73.7
	f=16	70.5	70.9	**70.7**	70.0	72.5	75.5	73.4	**76.4**	75.6	74.5

Table 5: Performance of different temporal aggregation strategies w/o temporal pooling. FLOPs and parameters of different models can be found in the supplementary material.

Temporal Aggregation. The essence of temporal modeling is how it aggregates temporal information. The 2D architecture offers great flexibility in temporal modeling. For example, TSM [38] and TAM [8] can be easily inserted into a CNN for learning spatio-temporal features. Here we analyze several basic temporal aggregations on top of the 2D architecture including 1D convolution (*Conv*, i.e., S3D [72]), 1D depthwise convolution (*dw Conv*, i.e., TAM), and TSM. We also consider the non-local network module (NLN) [69] for its ability to capture long-range temporal video dependencies add 3 NLN modules and 2 NLN modules at stage 2 and stage 3 of TSN-ResNet50, respectively as in [69].

Table 5 shows results of using different temporal aggregations as well as those of TSN (i.e., w/o any temporal aggregation) on InceptionV1 and ResNet50. The results suggest that *effective temporal modeling is required for achieving competitive results, even on datasets such as Kinetics where temporal information is thought as non-essential for recognition.* On the other hand, TAM and TSM, while being simple and efficient, demonstrate better performance than the I3D, 1D regular convolution and the NLN module, which have more parameters and FLOPs. We argue it is because the frames sampled under uniform sampling are sparse and it is not suitable to model temporal information in 3D convolution. While TAM and TSM use depthwise convolution that is more effective to model temporal information since it only consider the single feature map at different frames once instead of combining all channels of frames once. We also find the same pattern on full *Kinetics* in Table 3. Interestingly, the NLN module does not perform as expected on *Mini-SSV2*. This is possibly because NLN models temporal dependencies through matching spatial features between frames, which are weak in *Mini-SSV2*.

Locations of Temporal Modules. In [72] and [62], some preliminary analysis w.r.t the effect of the locations of temporal modules on 3D models was performed on Kinetics-400. In this experiment, we conduct a similar experiment on both *Mini-Kinetics* and *Mini-SSV2* to understand if this is so for 2D models. We modified TAM-ResNet18 in a number

# of TAMs	locations	Top-1 Acc. *Mini-SSV2*	*Mini-Kinetics*
8	All	59.1	69.08
4	Top-half	59.7	69.21
4	Bottom-half	56.5	69.27
4	Uniform-half	59.4	69.14

Top and bottom mean the residual blocks closer to output and input respectively.

Table 6: Performance comparison by using different numbers and locations of TAMs in ResNet18 (w/o temporal pooling).

of different ways by keeping: a) half of the temporal modules only in the bottom network layers (*Bottom-Half*); b) half of the temporal modules only in the top network layers (*Top-Half*); c) every other temporal module (*Uniform-Half*); and d) all the temporal modules (*All*). As observed in Table 6, only half of the temporal modules (*Top-Half*) is needed to achieve the best accuracy on *Mini-SSV2* while the accuracy on *Mini-Kinetics* is not sensitive to the number and locations of temporal modules. It is thus interesting to explore if this insightful observation can lead to an efficient but effective video architecture by mixing 2D and 3D modelings, similar to the idea of ECO in [79].

Disentangling Spatial and Temporal Effects. So far we have only looked at the overall spatio-temporal effects of a model (i.e., top-1 accuracy) in our analysis. Here we further disentangle the spatial and temporal contributions of a model to understand its ability of spatio-temporal modeling. Doing so provides great insights into which information, spatial or temporal, is more essential to recognition. We treat TSN w/o temporal pooling as the baseline spatial model as it does not model temporal information. TSN can evolve into different types of spatio-temporal models by adding temporal modules on top of it. With this, we compute the spatial and temporal contributions of a model as follows. Let $S_a^b(k)$ be the accuracy of a model of some architecture a that is based on a backbone b and takes k frames as input. For instance, $S_{I3D}^{ResNet50}(16)$ is the accuracy of a 16-frame I3D-ResNet50 model. Then the spatial contribution Φ_a^b and temporal improvement of a model Ψ_a^b (k is omitted here for clarity) are given by,

$$\begin{aligned}\Phi_a^b =& S_{TSN}^b / \max\left(S_a^b, S_{TSN}^b\right) \\ \Psi_a^b =& (S_a^b - S_{TSN}^b)/(100 - S_{TSN}^b).\end{aligned} \tag{1}$$

Note that Φ_a^b is between 0 and 1; $\Psi_a^b < 0$ indicates that temporal modeling is harmful to model performance. We further combine Φ_a^b and Ψ_a^b across all models with different backbone networks to obtain average spatial and temporal contributions of a network architecture, as shown below.

$$\bar{\Phi}_a = \frac{1}{Z_\Phi}\sum_{b\in B}\sum_{k\in K}\Phi_a^b(k), \quad \bar{\Psi}_a = \frac{1}{Z_\Psi}\sum_{b\in B}\sum_{k\in K}\Psi_a^b(k), \tag{2}$$

where B = {InceptionV1, ResNet18, ResNet50}, K = {8, 16, 32, 64}. Z_Φ and Z_Ψ are the normalization factors.

Datasets	Metrics	I3D	S3D	TAM
Mini-SSV2	$\bar{\Phi}_a$	0.53	0.53	0.52
	$\bar{\Psi}_a^{ta}$	0.46	0.45	0.47
	$\bar{\Psi}_a^{ta+tp}$	0.38	0.38	0.37
Mini-Kinetics	$\bar{\Phi}_a$	0.97	0.97	0.96
	$\bar{\Psi}_a^{ta}$	0.06	0.08	0.09
	$\bar{\Psi}_a^{ta+tp}$	-0.08	-0.10	-0.12
Mini-MiT	$\bar{\Phi}_a$	0.89	0.91	0.87
	$\bar{\Psi}_a^{ta}$	0.04	0.03	0.04
	$\bar{\Psi}_a^{ta+tp}$	0.02	0.02	0.04

$\bar{\Psi}_a^{ta}$: the improvement from temporal aggregation only.
$\bar{\Psi}_a^{ta+tp}$: the improvement from combining temporal

Table 7: Effects of spatio-temporal modeling.

Table 7 shows the results of $\bar{\Phi}_a$ and $\bar{\Psi}_a$ for three spatio-temporal representations. All three representations behave similarly, namely their spatial modeling contributes slightly more than temporal modeling on *Mini-SSV2*, much higher on *Mini-MiT*, and dominantly on Mini-Kinetics. This convincingly explains why a model lack of temporal modeling like TSN can perform well on *Mini-Kinetics*, but fail badly on *Mini-SSV2*. Note that similar observations have been made in the literature, but not in a quantitative way like ours. Furthermore, while all the approaches indicate the utmost importance of spatial modeling on mini-Kinetics, the results of $\bar{\Psi}_a^{ta}$ suggest that temporal modeling is more effective on *Mini-Kinetics* than on *Mini-MiT* for both 2D and 3D approaches. We also observe that temporal pooling deters the effectiveness of temporal modeling on all the approach from the results of $\bar{\Psi}_a^{ta+tp}$, which are constantly lower than $\bar{\Psi}_a^{ta}$. Such damage is especially substantial on *Mini-Kinetics*, indicated by the negative values of $\bar{\Psi}_a^{ta+tp}$.

7. Conclusion

In this paper, we conducted a comprehensive comparative analysis of several representative CNN-based video action recognition approaches with different backbones and temporal aggregations. Our extensive analysis enables better understanding of the differences and spatio-temporal effects of 2D-CNN and 3D-CNN approaches. It also provides significant insights with regard to the efficacy of spatio-temporal representations for action recognition.

Acknowledgments. This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via DOI/IBC contract number D17PC00341. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. This work is also supported by the MIT-IBM Watson AI Lab.

Disclaimer. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv:1609.08675*, 2016. 2

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 1, 2, 3, 4

[3] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *ICCV*, pages 3218–3226, 2015. 2

[4] 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. 1

[5] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M. Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *ECCV*, September 2018. 3

[6] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, June 2015. 2

[7] Linxi Fan, Shyamal Buch, Guanzhi Wang, Ryan Cao, Yuke Zhu, Juan Carlos Niebles, and Li Fei-Fei. RubiksNet: Learnable 3D-Shift for Efficient Video Action Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[8] Quanfu Fan, Chun-Fu (Ricarhd) Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More Is Less: Learning Efficient Video Representations by Temporal Aggregation Modules. In *NeurIPS*, 2019. 1, 2, 3, 4, 6, 7

[9] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, June 2020. 3, 5, 6

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv:1812.03982*, 2018. 1, 3, 5, 6

[11] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *NeurIPS*, pages 3468–3476, 2016. 2

[12] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *CVPR*, pages 4768–4777, 2017. 2

[13] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, pages 5378–5387, 2015. 2

[14] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, pages 12046–12055, 2019. 3

[15] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, pages 244–253, 2019. 2

[16] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, pages 971–980, 2017. 2

[17] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[18] Rohit Girdhar, Du Tran, Lorenzo Torresani, and Deva Ramanan. Distinit: Learning video representations without a single labeled video. *arXiv:1901.09244*, 2019. 3

[19] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, pages 759–768, 2015. 2

[20] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 1

[21] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCV*, pages 3154–3160, 2017. 3, 4

[22] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *CVPR*, June 2018. 1, 2, 3

[23] Dongliang He, Zhichao Zhou, Chuang Gan, Fu Li, Xiao Liu, Yandong Li, Limin Wang, and Shilei Wen. StNet: Local and Global Spatial-Temporal Modeling for Action Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8401–8408, July 2019. 3

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, June 2016. 1, 2

[25] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *CVPR*, pages 7366–7375, 2018. 3

[26] Noureldien Hussein, Efstratios Gavves, and Arnold W.M. Smeulders. Timeception for complex action recognition. In *CVPR*, June 2019. 3

[27] Matthew Hutchinson, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Chansup Byun, Micheal Houle, Matthew Hubbell, Micheal Jones, Jeremy Kepner, et al. Accuracy and performance comparison of video action recognition approaches. *arXiv:2008.09037*, 2020. 5, 7

[28] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE TPAMI*, 35(1):221–231, Jan 2013. 2

[29] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

[30] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 2

[31] 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:1705.06950*, 2017. 1, 2
[32] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 7
[33] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 3
[34] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion Feature Network: Fixed Motion Filter for Action Recognition. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 392–408, Cham, 2018. Springer International Publishing. 3
[35] Guy Lev, Gil Sadeh, Benjamin Klein, and Lior Wolf. Rnn fisher vectors for action recognition and image annotation. In *ECCV*, pages 833–850. Springer, 2016. 2
[36] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Collaborative Spatiotemporal Feature Learning for Video Action Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3, 6
[37] Xinyu Li, Bing Shuai, and Joseph Tighe. Directional Temporal Modeling for Action Recognition. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 275–291, Cham, 2020. Springer International Publishing. 3
[38] Ji Lin, Chuang Gan, and Song Han. Temporal Shift Module for Efficient Video Understanding. In *ICCV*, 2019. 1, 2, 3, 7
[39] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. TEINet: Towards an Efficient Architecture for Video Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11669–11676, Apr. 2020. 3
[40] Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *ICCV*, pages 5512–5521, 2019. 1
[41] Chenxu Luo and Alan L. Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3
[42] Brais Martinez, Davide Modolo, Yuanjun Xiong, and Joseph Tighe. Action recognition with spatial-temporal discriminative filter banks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3
[43] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *ICCV Workshops*, Oct 2019. 7
[44] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE TPAMI*, 2019. 1, 2, 7
[45] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, pages 4594–4602, 2016. 2
[46] Rameswar Panda and Amit K Roy-Chowdhury. Collaborative summarization of topic-related videos. In *CVPR*, pages 7083–7092, 2017. 2
[47] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, pages 563–579, 2018. 2
[48] Xiaojiang Peng, Changqing Zou, Yu Qiao, and Qiang Peng. Action recognition with stacked fisher vectors. In *ECCV*, pages 581–595. Springer, 2014. 2
[49] AJ Piergiovanni and Michael S. Ryoo. Representation flow for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3
[50] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, Oct 2017. 2, 3
[51] Michael S. Ryoo, AJ Piergiovanni, Mingxing Tan, and Anelia Angelova. Assemblenet: Searching for multi-stream neural connectivity in video architectures. In *International Conference on Learning Representations*, 2020. 3, 6
[52] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, pages 1049–1058, 2016. 2
[53] Gunnar A Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, pages 2137–2146, 2017. 3
[54] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 2
[55] Khurram Soomro, Amir Roshan Zamir, Mubarak Shah, Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv*, 2012. 7
[56] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, pages 843–852, 2015. 2
[57] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *CVPR*, pages 1102–1111, 2020. 1, 3
[58] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3
[59] C Szegedy, Wei Liu, Yangqing Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, and A Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 1
[60] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features With 3D Convolutional Networks. In *ICCV*, 2015. 2, 3
[61] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, October 2019. 2, 3, 6
[62] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *CVPR*, June 2018. 2, 3, 4, 7

[63] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV*, pages 4534–4542, 2015. 2

[64] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video Modeling With Correlation Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3, 6

[65] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, June 2018. 3

[66] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015. 2

[67] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, pages 4325–4334, 2017. 3

[68] 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 *ECCV*. Springer, 2016. 1, 2, 3, 4, 7

[69] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, June 2018. 2, 3, 7

[70] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, pages 3164–3172, 2015. 2

[71] Junwu Weng, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, Xudong Jiang, and Junsong Yuan. Temporal Distinct Representation Learning for Action Recognition. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 363–378, Cham, 2020. Springer International Publishing. 3

[72] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. In *ECCV*, Sept. 2018. 2, 3, 4, 7

[73] Zhongwen Xu, Yi Yang, and Alex G Hauptmann. A discriminative cnn video representation for event detection. In *CVPR*, pages 1798–1807, 2015. 2

[74] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[75] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015. 2

[76] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Trajectory convolution for action recognition. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 2208–2219, Red Hook, NY, USA, 2018. Curran Associates Inc. 3

[77] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, pages 803–818, 2018. 2, 3

[78] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Mict: Mixed 3d/2d convolutional tube for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[79] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *ECCV*, pages 695–712, 2018. 3, 8