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.

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 $1 \times 1$ 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 combina-
tion 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, it’s 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 stems 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).

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

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
5. Datasets, Training, Evaluation Protocols

To ensure fair comparison and facilitate reproducibility, 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 $i = 1 \ldots 4$. 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 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 $\text{module-backbone[-tp]}$ 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
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.

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 with and without 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 pooling 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.

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-
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, TAM, CorrNet, and X3D, 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 $3 \times 10$ clips for *Kinetics* and $3 \times 2$ 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 in 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

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 $3 \times 10$ clips on *Kinetics*, and $3 \times 2$ clips on *SSV2*, respectively.

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

### Table 2: Performance of SOTA models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Pretrain dataset</th>
<th>FLOPs</th>
<th>Dataset</th>
<th>SSV2</th>
<th>MiT</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3D-ResNet50</td>
<td>ImageNet</td>
<td>335.3G</td>
<td>76.61</td>
<td>62.84</td>
<td>31.21</td>
</tr>
<tr>
<td>I3D-ResNet50</td>
<td>None</td>
<td>335.3G</td>
<td>76.54</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TAM-ResNet50</td>
<td>ImageNet</td>
<td>171.5G</td>
<td>76.18</td>
<td>63.83</td>
<td>30.80</td>
</tr>
<tr>
<td>SlowFast-ResNet50-8-x8</td>
<td>[10]</td>
<td>None*</td>
<td>65.7G</td>
<td>76.40</td>
<td>60.10</td>
</tr>
<tr>
<td>SlowFast-ResNet101</td>
<td>ImageNet</td>
<td>654.7G</td>
<td>77.80</td>
<td>64.29</td>
<td>65.32</td>
</tr>
<tr>
<td>TAM-SlowFast-ResNet101</td>
<td>ImageNet</td>
<td>327.1G</td>
<td>77.61</td>
<td>65.52</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet50-8-x8</td>
<td>[10]</td>
<td>None*</td>
<td>65.7G</td>
<td>77.00</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet50-16-x8</td>
<td>[10]</td>
<td>None</td>
<td>124.5G</td>
<td>63.0</td>
<td>-</td>
</tr>
<tr>
<td>CorrNet-ResNet50</td>
<td>[64]</td>
<td>None*</td>
<td>115.9G</td>
<td>77.20</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet101-8-x8</td>
<td>[10]</td>
<td>None</td>
<td>125.9G</td>
<td>76.72</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet101-8-x8</td>
<td>[10]</td>
<td>None</td>
<td>125.9G</td>
<td>78.00</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet101-16-x8</td>
<td>[10]</td>
<td>None</td>
<td>213.9G</td>
<td>78.90</td>
<td>-</td>
</tr>
<tr>
<td>CSN-ResNet101</td>
<td>[61]</td>
<td>None*</td>
<td>183.9G</td>
<td>76.70</td>
<td>-</td>
</tr>
<tr>
<td>CorrNet-ResNet101</td>
<td>[64]</td>
<td>None*</td>
<td>224.9G</td>
<td>79.20</td>
<td>-</td>
</tr>
<tr>
<td>X3D-LJ</td>
<td>[9]</td>
<td>None*</td>
<td>24.8G</td>
<td>77.50</td>
<td>-</td>
</tr>
<tr>
<td>X3D-XL</td>
<td>[9]</td>
<td>None*</td>
<td>48.1G</td>
<td>79.10</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Pretrain dataset</th>
<th>FLOPs</th>
<th>Dataset</th>
<th>SSV2</th>
<th>MiT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SlowFast-ResNet101</td>
<td>ImageNet</td>
<td>327.1G</td>
<td>77.61</td>
<td>65.52</td>
<td>-</td>
</tr>
<tr>
<td>TAM-SlowFast-ResNet101</td>
<td>ImageNet</td>
<td>327.1G</td>
<td>77.61</td>
<td>65.52</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet50-8-x8</td>
<td>[10]</td>
<td>None*</td>
<td>65.7G</td>
<td>77.00</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet50-16-x8</td>
<td>[10]</td>
<td>None</td>
<td>124.5G</td>
<td>63.0</td>
<td>-</td>
</tr>
<tr>
<td>CorrNet-ResNet50</td>
<td>[64]</td>
<td>None*</td>
<td>115.9G</td>
<td>77.20</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet101-8-x8</td>
<td>[10]</td>
<td>None</td>
<td>125.9G</td>
<td>76.72</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet101-8-x8</td>
<td>[10]</td>
<td>None</td>
<td>125.9G</td>
<td>78.00</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast-ResNet101-16-x8</td>
<td>[10]</td>
<td>None</td>
<td>213.9G</td>
<td>78.90</td>
<td>-</td>
</tr>
<tr>
<td>CSN-ResNet101</td>
<td>[61]</td>
<td>None*</td>
<td>183.9G</td>
<td>76.70</td>
<td>-</td>
</tr>
<tr>
<td>CorrNet-ResNet101</td>
<td>[64]</td>
<td>None*</td>
<td>224.9G</td>
<td>79.20</td>
<td>-</td>
</tr>
<tr>
<td>X3D-LJ</td>
<td>[9]</td>
<td>None*</td>
<td>24.8G</td>
<td>77.50</td>
<td>-</td>
</tr>
<tr>
<td>X3D-XL</td>
<td>[9]</td>
<td>None*</td>
<td>48.1G</td>
<td>79.10</td>
<td>-</td>
</tr>
</tbody>
</table>

*These networks cannot be initialized from ImageNet due to its structure.

The authors of the paper.

Use RGB + Flow.

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Pretrain dataset</th>
<th>U-Sampling</th>
<th>D-Sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3D-ResNet50</td>
<td>ImageNet</td>
<td>76.07</td>
<td>76.61</td>
</tr>
<tr>
<td>TAM-ResNet50</td>
<td>ImageNet</td>
<td>76.45</td>
<td>76.18</td>
</tr>
<tr>
<td>SlowFast-ResNet50-8-x8</td>
<td></td>
<td>71.85</td>
<td>76.40</td>
</tr>
</tbody>
</table>

### 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 $\sim 5\%$ 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% $\sim 3\%$ on SSV2 in Table 2 in contrast to I3D and TAM.
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 vs. 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 efficiency in FLOPs. This clearly suggests that input sampling strategy of a model (i.e. uniform or dense) should factor in efficiency in FLOPs.

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 to 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 [58, 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.

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 TS-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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Frames</th>
<th>InceptionV1</th>
<th>ResNet50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mini-SSV2</td>
<td>f=8</td>
<td>70.4</td>
<td>74.1</td>
</tr>
<tr>
<td></td>
<td>f=16</td>
<td>70.5</td>
<td>74.1</td>
</tr>
<tr>
<td></td>
<td>f=8</td>
<td>70.8</td>
<td>74.1</td>
</tr>
<tr>
<td></td>
<td>f=16</td>
<td>70.9</td>
<td>74.1</td>
</tr>
</tbody>
</table>

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.
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 can evolve into different types of spatio-temporal models by adding temporal modules on top of it. For instance, \( S^b_{TSN} \) is the accuracy of a 16-frame I3D-ResNet50 model. Then the spatial contribution \( \Phi_a^b \) and temporal improvement of a model \( \Psi_a^b \) (\( k \) is omitted here for clarity) are given by,

\[
\Phi_a^b = S^b_{TSN} / \max (S^b_{a}, S^b_{TSN})
\]

\[
\Psi_a^b = (S^b_{a} - S^b_{TSN}) / (100 - S^b_{TSN}).
\]

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

\[
\Phi_a = \frac{1}{Z_{\Phi}} \sum_{b \in B} \sum_{k \in K} \Phi_a^b (k), \quad \Psi_a = \frac{1}{Z_{\Psi}} \sum_{b \in B} \sum_{k \in K} \Psi_a^b (k),
\]

where \( B = \{ \text{InceptionV1}, \text{ResNet18}, \text{ResNet50} \}, K = \{ 8, 16, 32, 64 \} \). \( Z_{\Phi} \) and \( Z_{\Psi} \) are the normalization factors.

Table 7 shows the results of \( \Phi_a \) and \( \Psi_a \) 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 \( \Psi_a^{tp} \) 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 \( \Psi_a^{tp} \), which are constantly lower than \( \Psi_a^{a} \). Such damage is especially substantial on Mini-Kinetics, indicated by the negative values of \( \Psi_a^{a+tp} \).

### 7. Conclusion

In this paper, we conducted a comprehensive comparative analysis of several representative CNN-based video action recognition approaches with different backbone 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

[31] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola,


[38] Ji Lin, Chuang Gan, and Song Han. Temporal Shift Module for Efficient Video Understanding. In ICCV, 2019. 1, 2, 3, 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


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


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


