An Image Classifier Can Suffice For Video Understanding

Quanfu Fan, Chun-Fu (Richard) Chen, Rameswar Panda
MIT-IBM Watson AI Lab, Cambridge, MA02412

Abstract

We propose a new perspective on video understanding by casting the video recognition problem as an image recognition task. We show that an image classifier alone can suffice for video understanding without temporal modeling. Our approach is simple and universal. It composes input frames into a super image to train an image classifier to fulfill the task of action recognition, in exactly the same way as classifying an image. We prove the viability of such an idea by demonstrating strong and promising performance on four public datasets including Kinetics400, Something-to-something (V2), MiT and Jester, using a recently developed vision transformer. We also experiment with the prevalent ResNet image classifiers in computer vision to further validate our idea. The results on Kinetics400 are comparable to some of the best-performed CNN approaches based on spatio-temporal modeling. Our code and models will be made available at https://github.com/IBM/sifar-pytorch.

1 Introduction

Video content is exploding on the internet. For example, YouTube is getting 720,000 hours of video uploaded daily, and it has become the second most popular search engine after Google.

The recent advances in convolutional neural networks (CNNs) \cite{19, 43}, along with the availability of large-scale video benchmark datasets \cite{22, 35, 9}, have significantly improved action recognition, one of the fundamental problems of video understanding. Many existing approaches for action recognition naturally extend or borrow ideas from image recognition. At the core of these approaches is spatio-temporal modeling, in which spatial information is processed by an image model (i.e., a backbone network) while time information is regarded as an additional dimension and fused jointly with space information by 3D CNNs \cite{45, 6, 16} or processed separately by efficient 2D CNNs \cite{26, 13}. CNN-based approaches have demonstrated strong capabilities in learning spatio-temporal feature representations from video.

Figure 1: Comparison of our proposed SIFAR (red) with SOTA approaches for action recognition on Kinetics400.

\footnote{https://www.oberlo.com/blog/youtube-statistics}

Preprint. Under review.
data. However a recent study surprisingly revealed that simple temporal modeling methods such as I3D \cite{carreira2017quo} and TAM \cite{kong2020temporal} perform comparably against more sophisticated spatio-temporal modeling approaches, and the recent progress in action recognition is largely attributed to the stronger backbone networks used in action models.

In this work we study video action recognition from a new perspective by casting the problem as an image recognition task. We aim to break the barrier between action recognition and image recognition and to show that the former can be directly addressed by the latter without temporal modeling. Our key idea is extremely straightforward. We first compose a sequence of input video frames into a super image (Fig. 2), which converts 3D spatio-temporal patterns in video into 2D spatial patterns. We then train an image classifier to fulfill the task of action recognition, in exactly the same way as classifying an image. For convenience, we dub our approach SIFAR, short for Super Image for Action Recognition.

Compared to spatio-temporal action modeling, SIFAR brings several advantages. Firstly, it is simple and universal. With one single line of code change in pytorch, SIFAR can repurpose any image classifier for action recognition. Secondly, by eliminating the need of temporal modeling in SIFAR, it makes action modeling easier, more computationally efficient and less parametric. Lastly, but not the least, the perspective of treating action recognition the same as image recognition unleash many possibilities for reusing existing techniques in a more mature image field to improve video understanding from various aspects. This includes better model architectures \cite{touvron2021training}, model pruning \cite{heo2021layer} and interpretability \cite{li2021towards}, to name a few.

We validate our idea by Swin Transformer \cite{liu2021 Swin}, a recently developed vision transformer that has demonstrated good performance on both image classification and object detection. By turning Swin Transformer Transformer into an action classifier as described above, we show that it produces strong performance on several benchmarks including Kinetics400 \cite{Kinetics400}, Something-to-something (V2) \cite{peng2019something}, MiT \cite{li2021masked} and Jester \cite{Jester} while being more efficient in both computation and parameters, compared to the SOTA spatio-temporal approaches for video action recognition. In addition, we further explore the potential of CNN-based classifiers directly used for action recognition under the proposed SIFAR framework. Surprisingly, they achieve very competitive results on Kinetics400 against existing CNN-based approaches that rely on much more sophisticated spatio-temporal modeling.

2 Related Work

Action Recognition from a Single Image. One direction for action recognition is purely based on a single image \cite{fu2019action, xie2020learning, xie2021video, liu2021single}. In \cite{fu2019action}, multiple small objects are first identified in a still image and then the target action is inferred from the relationship among the objects. E.g. if a hand holding a pen, the action is likely to be hand writing. Other approaches such as \cite{xie2020learning} propose to predict the missing temporal information in still images and then combine it with spatial information for action classification. There are also approaches such as motion-energy image (MEI) \cite{fu2019action} and Dynamic Image Network \cite{liu2021single} that attempt to summarize motion information in a video into a representation image for action classification. Nonetheless, our method does not attempt to understand a video from a single input image or a summarization image; instead it composites the video into a super image, and then classifies the image with an image classifier directly.

CNN-based Action Recognition. Action recognition is dominated by CNN-based models recently \cite{feichtenhofer2019slowfast, carreira2017quo, kong2020temporal, lin2019tsm, liu2021masked, Hara2019, v2cnn, heo2021layer, wang2021arnet}. These models process the video as a cube to extract spatial-temporal features via the proposed temporal modeling methods. E.g., SlowFast \cite{feichtenhofer2019slowfast} proposes two pathways whose speed is different to capture short-range and long-range time dependencies. TSM \cite{lin2019tsm} applies a temporal shifting module to exchange information between neighboring frames and TAM \cite{kong2020temporal} further enhances TSM by determining the amount of information to be shifted and blended. On the other hand, another thread of work attempts to select the key frame of an activity for faster recognition \cite{peng2021adaframe, heo2021layer, li2021towards}. E.g., Adaframe \cite{peng2021adaframe} employs a policy network to determine whether or not this is a key frame, and the main network only processes the key frames. ARNet \cite{li2021towards} determines what the image resolution should be used to save computations based on the importance of input frame images. Nonetheless, our approach is totally different from conventional action recognition. It simply uses an image classifier as a video classifier by laying out a video as a super image without explicitly modeling temporal information.
Figure 2: Overview of SIFAR. A sequence of input frames from a video are first made into a super image, which is then fed into a conventional image classifier for action recognition.

Action Recognition with Transformer. Following the vision transformer (ViT) [13], which demonstrates competitive performance against CNN models on image classification, many recent works attempt to extend the vision transformer for action recognition [36, 25, 3, 1, 14]. VTN [36], VidTr [25], TimeSformer [3] and ViViT [1] share the same concept that inserts a temporal modeling module into the existing ViT to enhance the features from the temporal direction. E.g., VTN [36] processes each frame independently and then uses a longformer to aggregate the features across frames. On the other hand, divided-space-time modeling in TimeSformer [4] inserts a temporal attention module into each transformer encoder for more fine-grained temporal interaction. MViT [14] develops a compact architecture based on the pyramid structure for action recognition. It further proposes a pooling-based attention to mix the tokens before computing the attention map so that the model can focus more on neighboring information. Nonetheless, our method is straightforward and applies the Swin [27] model to classify super images composed from input frames.

Note that the joint-space-time attention in TimeSformer [4] is a special case of our approach since their method can be considered as flattening all tokens into one plane and then performing self-attention over all tokens. Nonetheless, the memory complexity of such an approach is prohibitively high, and it is only applicable to image classifiers like ViT [13] with little inductive bias. On the other hand, our proposed SIFAR is general and applicable to any classifiers.

3 Approach

3.1 Overview of our Approach

The key insight of SIFAR is to turn spatio-temporal patterns in video data into purely 2D spatial patterns in images. Like their 3D counterparts, these 2D patterns may not be visible and recognizable by human. However, we hope they are characteristic of actions and identifiable by powerful neural network models. To that end, we make a sequence of input frame images from a video into a super image, as illustrated in Fig. 2, and then apply an image classifier to predict the label of the video. Note that the action patterns embedded in a super image can be complex and may involve both local (i.e. spatial information in a video frame) and global contexts (i.e. temporal dependencies across frames). It is thus understandable that to make our approach effective, an image classifier with strong capabilities in capturing short-range and long-range spatial dependencies in a super image is desired. For this reason, we explore the recently developed vision transformers based on self-attention to validate our proposed idea. These methods naturally own the ability to model global image contexts and have demonstrated competitive performance on image classification as well as action recognition against the best-performed CNN-based approaches.

Next we briefly describe Swin Transformer [27], an efficient and memory-friendly approach that we choose to implement our main idea in this work.

Preliminary. The Vision Transformer (ViT) [13] is a purely attention-based classifier borrowed from NLP. It consists of stacked transformer encoders, each of which is featured with a multi-head self-attention module (MSA) and a feed-forward network (FFN). While demonstrating promising results on image classification, ViT uses an isotropic structure and has a quadruple complexity w.r.t image resolution in terms of memory and computation. This significantly limits the application of ViT to many vision applications that requires high-resolution features such as object detection and segmentation. In light of this issue, several approaches [27, 8, 55] have been proposed to perform
region-level local self-attention to reduce memory usage and computation, and Swin Transformer is one of such improved vision transformers.

Swin Transformer first adopts a pyramid structure widely used in CNNs to reduce computation and memory. At the earlier layers, the network keeps high image resolution with fewer feature channels to learn fine-grained information. As the network goes deeper, it gradually reduces spatial resolution while expanding feature channels to model coarse-grained information. To further save memory, Swin Transformer limits self-attention to non-overlapping local windows (W-MSA) only. The communications between W-MSA blocks is achieved through shifting them in the succeeding transformer encoder. The shifted W-MSA is named as SW-MSA. Mathematically, the two consecutive blocks can be expressed as:

\[
y_k = \text{W-MSA}(\text{LN}(x_{k-1})) + x_{k-1},
\]

\[
x_k = \text{FFN}(\text{LN}(y_k)) + y_k,
\]

\[
y_{k+1} = \text{SW-MSA}(\text{LN}(x_k)) + x_k,
\]

\[
x_{k+1} = \text{FFN}(\text{LN}(y_{k+1})) + y_{k+1},
\]

(1)

where \(x_l\) is the features at the \(l^{th}\) layer and FFN and LN are feed-forward network and layer normalization, respectively.

In our case, SIFAR learns action patterns by sliding window, as illustrated in Fig 4. When the sliding window (blue box) is within a frame, spatial dependencies are learned. On the other hand, when the window (red box) spans across frames, temporal dependencies between them are effectively captured. The spatial pooling further ensures longer-range dependencies across frames captured.

**Creation of Super Image.** Given a set of frame images, we place them onto a regular grid (Fig. 3) to form a large super image. Different layouts give different spatial patterns for an action class. We hypothesize that a more compact structure such as a square grid may facilitate a model to learn temporal dependencies across frames as such a grid provides the shortest maximum distance between any two images. In our default setting, we use a \(3 \times 3\) layout for 8 images and a \(4 \times 4\) one for 16 images, respectively. Empty images are padded at the end if the grid is not fully filled up by the input images. There are other spatial arrangements for the input images (see Fig. 3 for more examples). However our experiments empirically show that a square grid performs the best.

**Implementation.** Once the spatial layout for the input frames is determined, implementing our idea in pytorch is as simple as inserting into an image classifier the following line of code:

```python
# create a super image with a layout (sh, sw) pre-specified by the user.
x = rearrange(x, 'b c (sh sw) h w -> b c (sh h) (sw w)', sh=sh, sw=sw)
```

The trivial code change described above transforms an image classifier into an video action classifier. Our experiments show that the same training and evaluation protocols for action models can be still applied to the repurposed image classifier without change.

**Sliding Window.** As previously mentioned, Swin Transformer performs self-attention within a small local window to save memory. It uses a uniform window size across all layers, and the default window size is 7 in the original paper. Since a larger window leads to more interactions across frames and is beneficial for SIFAR to learn long-range temporal dependencies in super images, we slightly modify the architecture of Swin-B for it to take different window sizes in self-attention. In particular, we keep the same window size for all the layers except the last one, whose window is as large as its image resolution, implying a global self-attention including all the tokens. Since the last layer of Swin-B has only two transformer encoders, the computational overhead imposed by an increased window size is quite small while the memory usage is still manageable, as indicated in Table 1.

---

**Figure 3: Grid Layout.** We apply a grid to lay out the input frames. Illustrated here are several possible layouts for 8 frames, i.e., a) \(1 \times 8\), b) and c) \(2 \times 4\), and d) \(3 \times 3\), respectively. Empty images are padded at the end if the grid is not fully filled up by the input frames.

![Grid Layout](image)

---

Our idea in pytorch is as simple as inserting into which changes the input of a video to a super image.

---

---

---

The change of window size may result in adjustment of the input image size as the image resolution at each layer must be divisible by the window size in Swin Transformer. As noted in Table 1, SIFAR-7 keeps the vanilla architecture of Swin-B. SIFAR-12 is more efficient than SIFAR-7 because SIFAR-12 takes images of a smaller size ($192^2$) as input. We demonstrate later in Section 4.1 that a larger window is critical for SIFAR to achieve good performance on more temporal datasets such as SSV2.

Figure 4: Swin Transformer does self-attention in a local window. In SIFAR, when the window (blue box) is within a frame, spatial dependencies are learned within a super image (4 frames here). When the window spans across different frames (red box), temporal dependencies between them are effectively captured. The spatial pooling further ensures longer-range dependencies to be learnt.

4 Experiments

4.1 Datasets and Experimental Setup

Datasets. We use Kinetics400 (K400) [22], Something-Something V2 (SSV2) [18], Moments-in-time (MiT) [35] and Jester [33] in our evaluation. Kinetics400 is a widely-used benchmark for action recognition, which includes ~240k training videos and 20k validation videos in 400 classes. SSV2 contains 220k videos of 174 types of predefined human-object interactions with everyday objects. This dataset is known for its high temporal dynamics. MiT is a fairly large collection of one million 3-second labeled video clips, involving actions not only from humans, but also from animals, objects and natural phenomena. The dataset includes around 800k training videos and 33,900 validation videos in 339 classes. Jester contains actions of predefined hand gestures, with 118,562 and 14,787 training and validation videos over 27 classes, respectively.

Training. We employ uniform sampling to generate video input for our models. We train all our models by fine-tuning a Swin-B model [27] pretrained on ImageNet-21K [11], except for those 16-frame SSV2 models in Table 2 which are initialized from the corresponding Kinetics400 models in Table 3. Our training recipe and augmentations closely follow [44]. First, we apply multi-scale jitter to augment the input [47] with different scales and then randomly crop a target input size (e.g. $8 \times 224 \times 224$ for SIFAR-7). We then use Mixup [54] and CutMix [52] to augment the data further, with their values set to 0.8 and 1.0, respectively. After that, we rearrange the image crops as a super image. Furthermore, we apply drop path [43] with a rate of 0.1, and enable label smoothing [42] at a rate of 0.1.

All our models were trained using V100 GPU cards with 16G or 32G memory. Depending on the size of a model, we use a batch size of 96, 144 or 192 to train the model for 15 epochs on MiT or 30 epochs on other datasets, including 5 warming-up epochs. The optimizer used in our training is AdamW [31] with a weight decay of 0.05, and the scheduler is Cosine [30] with a base linear learning rate of 0.0001.

Inference. We first scale the shorter side of an image to the model input size and then take three crops (top-left, center and bottom-right) for evaluation. The average of the three predictions is used as the final prediction. We report results by top-1 and top-5 classification accuracy (%) on validation data, the total computational cost in FLOPs and the model size in MB.

4.2 Main Results

Comparison with Baselines. We first compare our approaches with several representative CNN-based methods including I3D [6], TSM [26] and TAM [15]. We also include two other models based on TimeSformer [4] but using the same backbone Swin-B [27] as our models. All the models in comparison take 8 frames as input. As can be seen from Table 2 our approach achieves better performance than previous methods on all datasets.
Table 3: Comparison with Other Approaches on Kinetics400.

| Model              | #Frames | Pretrain | Params(M) | FLOPs(G) | Top-1 | Top-5 |
|--------------------|---------|----------|-----------|----------|-------|-------|
| TSN-R50 [47]       | 32      | IN-1K    | 24.3      | 170.8 x 30 | 69.8  | 89.1  |
| TAM-R50 [15]       | 32      | IN-1K    | 24.4      | 171.5 x 30 | 76.2  | 92.6  |
| I3D-R50 [6]        | 32      | IN-1K    | 47.0      | 335.3 x 30 | 76.6  | 92.7  |
| I3D-R50+NL [48]    | 32      | IN-1K    | -         | 282 x 30   | 76.5  | 92.6  |
| I3D-R101+NL [48]   | 32      | IN-1K    | -         | 359 x 30   | 77.7  | 93.3  |
| ip-CSN-152 [46]    | 32      | -        | 32.8      | 109 x 30   | 78.9  | 92.8  |
| SlowFast8x8 [17]   | 32      | -        | 27.8      | 65.7 x 30  | 77.0  | 92.6  |
| SlowFast8x8+NL [17] | 64     | -        | 59.9      | 116 x 30   | 78.3  | 93.5  |
| X3D-M [16]         | 16      | -        | 3.8       | 6.2 x 30   | 76.0  | 92.3  |
| X3D-XL [16]        | 16      | -        | 11.0      | 48.4 x 30  | 79.1  | 93.9  |
| TimeSformer [3]    | 8       | IN-21K   | -         | 121.4 x 3 | 78.0  | -     |
| TimeSformer-HR [3] | 16      | IN-21K   | 121.4     | 1703 x 3   | 79.7  | -     |
| TimeSformer-L [3]  | 96      | IN-21K   | 121.4     | 2380 x 3   | 80.7  | -     |
| ViViT-L [1]        | 32      | IN-21K   | 310.8     | 3992 x 12  | 81.3  | 94.7  |
| MViT-B [14]        | 16      | -        | 36.6      | 70.5 x 5   | 78.4  | 93.5  |
| MViT-B [14]        | 64      | -        | 36.6      | 455 x 9    | 81.2  | 95.1  |
| SIFAR-12           | 8       | IN-21K   | 87        | 106 x 3    | 80.0  | 94.5  |
| SIFAR-14           | 8       | IN-21K   | 87        | 147 x 3    | 80.4  | 94.4  |
| SIFAR-12‡          | 16      | IN-21K   | 87        | 189 x 3    | 80.4  | 94.4  |
| SIFAR-14‡          | 16      | IN-21K   | 87        | 263 x 3    | 81.1  | 94.6  |
| SIFAR-15‡          | 8       | IN-21K   | 87        | 303 x 3    | 81.1  | 94.6  |
| SIFAR-12‡          | 8       | IN-21K   | 87        | 423 x 3    | 81.6  | 95.2  |

Our approach substantially outperforms the CNN baselines on Kinetics400 while achieving comparable results on SSV2. Our approach is also better than TimeSformer on both datasets. These results clearly demonstrate that an image classifier can learn expressive spatio-temporal patterns effectively from super images for action recognition. In another word, an image classifier can suffice for video understanding without explicit temporal modeling.

The results also confirm that a larger sliding window is more helpful capturing temporal dependencies on temporal datasets like SSV2. Our architecture design allows for global self-attention only in the last layer of a model (see Table 1). This substantially mitigates the memory issue in training vision transformers.

Kinetics400. We report our Kinetics400 results in Table 3 and compare them with SOTA approaches. Our 8-frame models (SIFAR-12 and SIFAR-14) achieve 80.0% and 80.4% top-1 accuracies, outperforming all the CNN-based approaches while being more efficient than the majority of them. SIFAR-15‡ and SIFAR-12‡ further gain ~ 1% improvement, benefiting from higher input image resolutions. Especially, SIFAR-12‡ yields an accuracy of 81.6%, the best among all the very recently developed approaches based on vision transformers including TimeSformer [3] and MViT-B [14]. Our approach also offers clear advantages in terms of FLOPs and model parameters compared to other approaches except MViT-B. For example, SIFAR-12‡ has 5× and 37× fewer FLOPs than TimeSformer-L and ViViT-L, respectively, while being 1.4× and 3.6× smaller in model size.

SSV2. Table 4 lists the results of our models and SOTA approaches on SSV2. There is a 1.0 ~ 2.5% performance gap between our approach and the best-performed CNN methods. However, our approach performs on par with other transformer-based method such as TimeSformer [4] and VidTr-L [25] under a similar setting. Note that ViViT-L [1] achieves better results with a larger and stronger backbone ViT-L [13]. MViT-B [14] is an efficient multi-scale architecture, which can process much longer input sequences to capture fine-grained motion patterns in SSV2 data. Although Swin Transformer designed to be memory-friendly, it still remains computationally challenging for such an approach to handle an input sequence more than 16 frames, especially
Table 4: Comparison with Other Approaches on SSV2.

| Model               | #Frames | Params(M) | FLOPs(G) | Top-1  | Top-5  |
|---------------------|---------|-----------|----------|--------|--------|
| TAM-R50 [15]        | 32      | 24.4      | 171.5×6  | 63.8   | 88.3   |
| I3D-R50 [6]         | 32      | 47.0      | 335.3×6  | 62.8   | 88.0   |
| TSM-R50 [26]        | 16      | 24.3      | 65×6     | 63.4   | 88.5   |
| TAM-blR101 [15]     | 64      | 40.2      | 96.4×1   | 65.2   | 90.3   |
| MSNet [23]          | 16      | 24.0      | 67×1     | 64.7   | 89.4   |
| STM [21]            | 16      | 24.0      | 67×30    | 64.2   | 89.8   |
| TEA [28]            | 16      | —         | 70×30    | 65.1   | 89.9   |
| TimeSformer [3]     | 8       | 121.4     | 196×3    | 59.5   | —      |
| TimeSformer-HR [3]  | 16      | 121.4     | 1703×3   | 62.5   | —      |
| ViViT-L [11]        | 32      | 100.7     | —        | 65.4   | 89.8   |
| VidTr-L [25]        | 32      | —         | —        | 60.2   | —      |
| MViT-B [14]         | 16      | 36.6      | 70.5×3   | 64.7   | 89.2   |
| MViT-B [14]         | 64      | 36.6      | 455×3    | —      | —      |
| SIFAR-12 †          | 8       | 87        | 106×3    | 60.1   | 86.8   |
| SIFAR-14 †          | 8       | 87        | 147×3    | 60.6   | 86.7   |
| SIFAR-12 ‡          | 16      | 87        | 189×3    | 61.9   | 87.4   |
| SIFAR-14 ‡          | 16      | 87        | 263×3    | 62.6   | 88.3   |

Table 5: Comparison with Other Approaches on MiT and Jester.

(a) MiT.

| Model            | Top-1  | Top-5  |
|------------------|--------|--------|
| TRN-Inception [57]| 28.3   | 53.9   |
| TAM-R50 [15]     | 30.8   | 58.2   |
| I3D-R50 [7]      | 31.2   | 58.9   |
| SlowFast-R50-8×8 [17]| 31.2 | 58.7   |
| CoST-R101 [24]   | 32.4   | 60.0   |
| SRTG-R3D-101 [41]| 33.6   | 58.5   |
| AssembleNet [37] | 33.9   | 60.9   |
| ViViT-L [11]     | 38.0   | 64.9   |
| SIFAR-15 †       | 38.5   | 67.4   |
| SIFAR-12 ‡       | **39.9** | **69.2** |

(b) Jester.

| Model            | Top-1  | Top-5  |
|------------------|--------|--------|
| TSN-Inception [42] | 95.0   | 99.9   |
| TRN-Inception [57]| 95.3   | —      |
| TSM-R50 [26]     | 95.0   | 99.9   |
| PAN-R50 [53]     | 99.6   | 99.8   |
| STM-R50 [20]     | 96.7   | 99.9   |
| I3D-R50 [16]     | 96.4   | —      |
| TAM-R50 [15]     | 96.4   | —      |
| SlowFast-R50-8×8 [17]| 96.8 | —      |
| SIFAR-12 †       | 97.2   | 99.9   |
| SIFAR-14 ‡       | **97.2** | **99.9** |

when a larger sliding window size is desired for self-attention such as in the case of SIFAR-15 ‡ and SIFAR-12 ‡. Our results suggest that developing more efficient architectures of vision transformer is an area of improvement and future work for SIFAR to take advantage of more input frames.

**MiT and Jester.** We further evaluate our approach on another two popular benchmarks: MiT [35] and Jester [33]. Here we only consider the best single models from other approaches for fair comparison. As shown in Table 5 our proposed SIFAR achieves competitive results again on both datasets, surpassing all other models in comparison. Note that MiT is a large diverse dataset containing label noise. However, SIFAR-12 ‡ is ~2% better than ViViT-L [11] based on a computationally expensive backbone, and significantly outperforms AssembleNet [38] based on neural architecture search.

**Classification by CNNs.** We also test our proposed approach using the ResNet image classifiers on both SSV2 and Kinetics400 datasets. For fairness, the ResNet models are pretrained on ImageNet-21K. Table 6 shows the results. As seen from Table 6 our models clearly outperform CNN-based models for action recognition on Kinetics400. Especially, with a strong backbone R152x2 (a model 2× wider than Resnet152), SIFAR-R152x2 achieves a superior accuracy of 80.0%, which is surprisingly better than the best CNN results (Slow-Fast16x8+NL: 79.8%) reported in Table 3 On SSV2, our models are not as good as the CNN models but very competitive with the self-attention model (SIFAR-7) under the common experiment setting. Our results shows that stronger image classifiers tend to result in better accuracy. How to develop better CNN based SIFAR to close the gap with respect to spatio-temporal methods on temporal data such as SSV2 is an interesting direction for future work.

7
Table 6: SIFAR based on CNN. The base denotes that Mixup, CutMix and drop-path are disabled; the Aug. follows our training setting. All results are evaluated under three crops.

| Model       | # Frames | SSV2 Base | SSV2 Aug | Kinetics400 Base | Kinetics400 Aug |
|-------------|----------|-----------|----------|------------------|-----------------|
| I3D-R50     | 8        | 61.1      | 72.6     | -                | -               |
| TSM-R50     | 8        | 59.1      | 74.1     | -                | -               |
| TAM-R50     | 8        | 62.0      | 72.2     | -                | -               |
| SIFAR-R50   | 8        | 48.6      | 50.8     | 73.5             | 73.2            |
| SIFAR-R50   | 16       | 46.8      | 49.0     | 73.3             | 73.5            |
| SIFAR-R101  | 8        | 53.7      | 56.3     | 75.5             | 76.6            |
| SIFAR-R101  | 16       | 52.5      | 56.2     | 75.8             | 77.4            |
| SIFAR-R152  | 8        | 53.5      | 58.2     | 76.0             | 79.0            |
| SIFAR-R152  | 16       | 54.5      | 58.6     | 77.8             | 80.0            |
| SIFAR-7     | 8        | 53.4      | 56.7     | 78.7             | 79.6            |

*: a model two times wider than R152

Table 7: Ablation Study. The effects of each component on model accuracy.

(a) Super Image Layout. (SIFAR-12 models on SSV2)

| Layout       | Top-1   | Top-5   |
|--------------|---------|---------|
| 1×8 (Fig. 3a)| 44.4    | 74.4    |
| 2×4 (Fig. 3b)| 58.6    | 85.5    |
| 2×4 (Fig. 3c)| 58.1    | 85.1    |
| 3×3 (Fig. 3d)| 60.1    | 86.8    |

(b) Absolute Positioning Embedding.

| Model       | SSV2 w/ APE | SSV2 w/o APE | Kinetics400 w/ APE | Kinetics400 w/o APE |
|-------------|-------------|---------------|--------------------|---------------------|
| SIFAR-7     | 56.6        | 56.4          | 79.7               | 79.6                |
| SIFAR-12    | 60.1        | 59.5          | 79.7               | 80.0                |
| SIFAR-14    | 60.6        | 60.1          | 80.4               | 80.4                |

It is also noticed that as opposed to CNN based spatio-temporal models, SIFAR models based on CNN classifiers do not benefit much from more input frames. More frames are even harmful for less deep models such as ResNet50. We believe this is because super images composed from more frames have larger spatial resolutions, implying that stronger capabilities from a classifier are needed to capture longer-range dependencies in the data. This also explains why vision transformers based on self-attention performs substantially better than CNN classifiers on SSV2.

4.3 Ablation Studies

In this section, we conduct ablation studies to provide more insights about our approach.

How does an image layout affect the performance? The layout of a super image determines how spatio-temporal patterns are embedded in it. We hypothesize that the layout could affect the learning effectiveness of a model. To analyze this, we trained a SIFAR model on SSV2 for each layout illustrated in Fig. 3. As shown in Table 7(a) a strip layout performs the worst while a grid layout produces the best results, which confirms our hypothesis.

Does absolute positioning embedding help? Absolute Position Embedding (APE) assigns fixed or learnable position information to each token in a transformer model, and it has been proven helpful to vision transformers such as ViT. However, the Swin paper shows that when relative position bias are added, APE is only moderately beneficial for classification, but not for object detection and segmentation. They thus conclude that inductive bias that encourages certain translation invariance is still important for vision tasks. To find out whether or not APE is effective in our approach, we add APE to each frame rather than each token. The results in Table 7(b) indicate that APE slightly improves model accuracy on SSV2, but is harmful to Kinetics400. In our main results, we thus apply APE to SSV2 only.

What does SIFAR learn? One big advantage of our proposed approach is that many techniques developed in the image domain now can be directly used for video understanding without re-invention. Here we apply ablation cam, an image model interpretability technique, to understand what our models learn. Fig. 5 shows the Class Activation Maps (CAM) of 4 actions correctly predicted by SIFAR-12. Not surprisingly, the model learns to attend to objects relevant to the target action such as the hula hoop in a) and soccer ball in b). In c) and d), the model seems to correctly focus on where meaningful motion happens.
We have presented a new perspective for action recognition by casting the problem as an image recognition task. Our idea is simple and universe, and with one line of code to transform an sequence of input frames into a super image, it can re-purpose any image classifier for action recognition. We have implemented our idea with both CNN-based and transformer-based image classifiers, both of which show promising results on several popular public video benchmarks. Our experiments and results show that applying super images for video understanding is an interesting direction worth further exploration.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021.

[2] Y Alp Aslandogan and Clement T. Yu. Techniques and systems for image and video retrieval. IEEE transactions on Knowledge and Data Engineering, 11(1):56–63, 1999.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding?, 2021.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is Space-Time Attention All You Need for Video Understanding? arXiv.org, February 2021.

[5] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3034–3042, 2016.

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

[7] Chun-Fu Chen, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, and Quanfu Fan. Deep analysis of cnn-based spatio-temporal representations for action recognition, June 2021.

[8] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting Spatial Attention Design in Vision Transformers. arXiv.org, April 2021.

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. CoRR, abs/2006.13256, 2020.

[10] James Davis and Aaron Bobick. The representation and recognition of action using temporal templates. In Proceedings of the IEEE International Conference on Computer Vision, pages 2736–2744, 1997.

[11] 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.

[12] Saurabh Desai and Harish G. Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 972–980, 2020.
[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021.

[14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers, 2021.

[15] Quanfu Fan, Chun-Fu (Ricardh) Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More Is Less: Learning Efficient Video Representations by Temporal Aggregation Modules. In NeurIPS, 2019.

[16] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In CVPR, June 2020.

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. arXiv:1812.03982, 2018.

[18] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Frueind, Peter Yianilos, Mortiz Mueller-Freitag, et al. The “something something” video database for learning and evaluating visual common sense. In ICCV, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[20] 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.

[21] Boyuan Jiang, Mengmeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 2000–2009, 2019.

[22] 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.

[23] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In ECCV, 2020.

[24] 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.

[25] Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. VidTr: Video Transformer Without Convolutions. arXiv.org, 2021.

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

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv.org, March 2021.

[28] Zhaoyang Liou, Donghao Luo, Yahiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. TE1Net: Towards an Efficient Architecture for Video Recognition. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07):11669–11676, April 2020.

[29] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE International Conference on Computer Vision, pages 2736–2744, 2017.

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In International Conference on Learning Representations, 2017.

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In International Conference on Learning Representations, 2019.

[32] Yu-Fei Ma, Lie Lu, Hong-Jiang Zhang, and Mingjing Li. A user attention model for video summarization. In Proceedings of the tenth ACM international conference on Multimedia, pages 533–542, 2002.

[33] 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.

[34] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and Rogerio Feris. Ar-net: Adaptive frame resolution for efficient action recognition. arXiv preprint arXiv:2007.15796, 2020.

[35] 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.
[36] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network, 2021.
[37] Michael S. Ryoo, A. J. Piergiovanni, Mingxing Tan, and Anelia Angelova. AssemblerNet: Searching for multi-stream neural connectivity in video architectures. *CoRR*, abs/1905.13209, 2019.
[38] Michael S. Ryoo, AJ Piergiovanni, Mingxing Tan, and Anelia Angelova. AssemblerNet: Searching for multi-stream neural connectivity in video architectures. In *International Conference on Learning Representations*, 2020.
[39] Marjaneh Safaei, Pooyan Balouchian, and Hassan Foroosh. Ticnn: A hierarchical deep learning framework for still image action recognition using temporal image prediction. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3463–3467, 2018.
[40] Marjaneh Safaei and Hassan Foroosh. Still image action recognition by predicting spatial-temporal pixel evolution. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 111–120, 2019.
[41] Alexandros Stergiou and Ronald Poppe. Learn to cycle: Time-consistent feature discovery for action recognition. *arXiv*, 2020.
[42] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
[43] Mingxing Tan and Quoc Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114, Long Beach, California, USA, June 2019. PMLR.
[44] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.
[45] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features With 3D Convolutional Networks. In *ICCV*, 2015.
[46] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, October 2019.
[47] Limin Wang, Yujun 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.
[48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, June 2018.
[49] Zuxuan Wu, Caiming Xiong, Yu-Gang Jiang, and Larry S Davis. LiteEval: A Coarse-to-Fine Framework for Resource Efficient Video Recognition. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
[50] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S. Davis. AdaFrame: Adaptive frame selection for fast video recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1278–1287, 2019.
[51] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
[52] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
[53] Can Zhang, Yuexian Zou, Guang Chen, and Lei Gan. PAN: Towards Fast Action Recognition via Learning Persistence of Appearance. *arXiv*, 2020.
[54] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.
[55] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-Scale Vision Longformer: A New Vision Transformer for High-Resolution Image Encoding. *arXiv.org*, March 2021.
[56] Zhichen Zhao, Huimin Ma, and Shaodi You. Single image action recognition using semantic body part actions. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
[57] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, pages 803–818, 2018.