Recurring the Transformer for Video Action Recognition

Jiewen Yang[1] Xingbo Dong[1,2*] Liujun Liu[1*] Chao Zhang[1]
Jiajun Shen[1] Dahai Yu[1]
[1]TCL Corporate Research (HK) Co., Ltd, [2]Yonsei University, Seoul, South Korea
{jiewen.yang,liujun.liu,chao46.zhang,sjj,dahai.yu}@tcl.com, xingbo.dong@yonsei.ac.kr

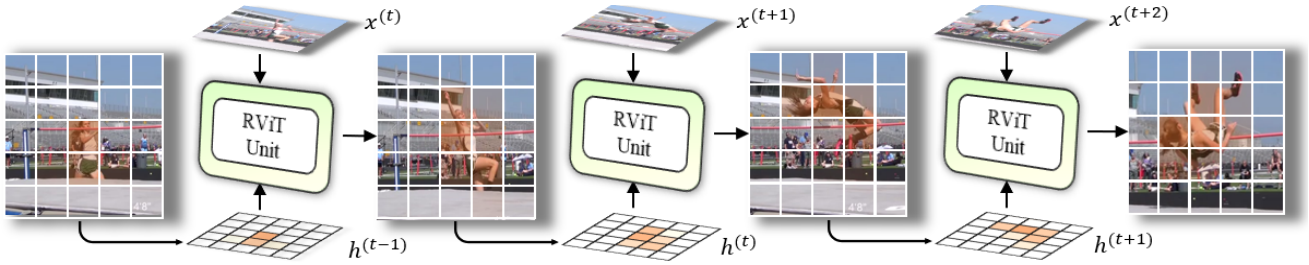


Figure 1. **The overview of the recurrent transformer working pipeline.** The input frame $x^{(t)}$ and hidden state $h^{(t-1)}$ jointly determine the current hidden state $h^{(t)}$ and output $O^{(t)}$. The hidden state contains attention information which can be transferred to next frame.

Abstract

Existing video understanding approaches, such as 3D convolutional neural networks and Transformer-Based methods, usually process the videos in a clip-wise manner; hence huge GPU memory is needed and fixed-length video clips are usually required. To alleviate those issues, we introduce a novel ***Recurrent Vision Transformer (RViT)*** *framework based on spatial-temporal representation learning to achieve the video action recognition task. Specifically, the proposed RViT is equipped with an* ***attention gate*** *to build interaction between current frame input and previous hidden state, thus aggregating the global level inter-frame features through the hidden state temporally. RViT is executed recurrently to process a video by giving the current frame and previous hidden state. The RViT can capture both spatial and temporal features because of the attention gate and recurrent execution. Besides, the proposed RViT can work on variant-length video clips properly without requiring large GPU memory thanks to the frame by frame processing flow. Our experiment results demonstrate that RViT can achieve state-of-the-art performance on various datasets for the video recognition task. Specifically, RViT can achieve a top-1 accuracy of* 81.5% *on Kinetics-400,* 92.31% *on Jester,* 67.9% *on Something-Something-V2, and an mAP accuracy of* 66.1% *on Charades.*

*Work done while interning at TCL Corporate Research (HK) Co., Ltd. and equal contribution

1. Introduction

Existing video understanding works, such as [6, 17, 18, 21, 27, 57], usually utilize the 3D-CNNs network to achieve the spatial-temporal features extraction. With the successful adaptation of the Visual Transformer [11] for vision tasks, Transformer-based methods become a hot topic for video understanding tasks. TimeSformer [3], ViViT [1], VTN [40], Mformer [41] and MViT [15] are typical representatives. Though Transformer-based methods on computer vision task can achieve significant performance, tremendous computing memory is needed for those methods, which hinder the deployment of such schemes.

On the other hand, some studies [23, 42, 59] show that human visual attention in a video is driven by prior knowledge. For example, [59] points out that human often focuses on user-interested regions of videos, and prominent actions draw more attention than their surrounding neighbours at the initial sight. In the video understanding pipeline of humans, information from the previous frame usually can help determine the attention in the subsequent frame of the video in a recurrent manner.

From the human attention perspective, we reason that there are two categories of information contained in a video: (i) spatial (single frame), (ii) temporal (inter-frames). Both the spatial feature in the current frame and the temporal feature aggregated from prior frames play a crucial role in video understanding tasks. Meanwhile, the temporal features from adjacent frames usually show high similarity.

However, existing clip-wise approaches generally extract the temporal features at each processing batch, which leads to non-interest information included in the temporal features. Additionally, temporal features are usually extracted from a fix-length clip instead of a length-adaptive clip.

Motivated by the above discussion, we proposed a novel recurrent processing pipeline, namely Recurrent Vision-Transformer (RViT), to achieve the video action recognition task in this work. Specifically, the proposed RViT framework is based on an attention gate enabled RViT unit, given the current input frame $x^{(t)}$ and hidden state $h^{(t-1)}$ from the previous frame, an output $O^{(t)}$ and a hidden state $h^{(t)}$ will be generated from the current RViT unit. To achieve an length-adaptive temporal feature extraction, an attention gate is designed to transfer the temporal (inter-frames) feature through the hidden state instead of extracting temporal feature from every frame by batch.The aggregated temporal features through the hidden state is utilized to attend the spatial features in the subsequent processing flow.

Our contributions are: (i) An end-to-end Recurrent Vision-Transformer is proposed to process video sequences for action recognition. The proposed model consumes less GPU memory thanks to the frame-flow processing and achieves state-of-the-art performance simultaneously; (ii) A novel attention gate is incorporated into the RViT unit to preserve inter-frame attention information through the hidden state; Thus, an interaction between the aggregated temporal features and the current spatial feature can be established. (iii) Our extensive experiments demonstrate that state-of-the-art performance can be achieved for action recognition task. Our method can achieve a top-1 accuracy of 81.5% on Kinetics-400, 92.31% on Jester, 67.9% on Something-Something-V2, and an mAP accuracy of 66.1% on Charades. Additionally, temporal attention has been demonstrated visually.

2. Related Work

Convolution-Based Method and Self-Attention Convolution neural network (CNN) have achieved remarkable performance in computer vision tasks [22, 32, 35, 47–49]. For video understanding task, CNN methods can be generally categorised into two types, (i) extend the 2D CNN model in temporal dimension by using two stream network [6, 18, 19, 46, 51]; (ii) 3D convolution [16, 21, 28, 33, 43, 54].

The hybrids of self-attention and CNNs have demonstrated great success on image and video tasks. For example, the Non-Local Network [54] employs an attention method similar to the self-attention from transformer [52] to achieve the vision task.

Vision Transformer Transformer was originally proposed for Nature Language Processing tasks [9, 52]. Recently, transformer-based networks have also been adopted on computer vision tasks. For example, a transformer-based network is designed in the DETR [5] for object detection by combining the convolutional feature maps. Dosovitskiy et al. [11] proposed the Visual Transformer (ViT) and demonstrated that transformer framework without convolution layer can also achieve good performance on image processing tasks. Transformer-based models have also been adopted for video tasks [1, 3, 15, 40, 41]. Specifically, the ViViT [1] uses two transformer encoders to process spatial and temporal information, respectively. The TimeSformer [3] is a convolution-free approach that expands the spatial-only self-attention to joint spatial-temporal attention. The VTN [40] uses a 2D spatial feature extraction model based on a temporal-attention-based encoder to build an efficient architecture for video understanding. The MViT [15] proposed multi-head pooling attention with the specific spatial-temporal resolution and achieved encouraging performance.

Most existing transformer-based methods are designed in a parallel processing manner to process a batch of frames at once on video tasks. Such methods usually require a large GPU memory, and the temporal features are extracted within the batch, hence limited information is contained in the temporal features.

To address the discussed problems, the recursive method might be a good option. The usage of recursive methods has been demonstrated successfully for video tasks, for example, ConvLSTM [58] and ConvGRU [2]. Besides, indicated by some research works [8, 29, 34], the transformer mechanism shows similarity to RNN. The parameter sharing between transformer blocks can also lead to better performance [8, 34]. On the other hand, self-attention designed for video tasks suggests that the transformer-based method can establish the interaction between spatial and temporal domain [1, 3, 15, 40].

Based on the above literature review, we notice that no existing approach adopts a recursive mechanism into the transformer to achieve the video action recognition task. The usage of the recursive mechanism may benefit the performance and alleviate the expensive GPU memory cost. We propose the RViT framework to process the variant-length video clips via recurring the standard ViT design to address the above issues.

3. Recurrent-Based Transformer

Previous researches [4, 36, 59] indicate that the Human Visual System (HVS) has the ability to orient attention to the most informative area of visual scenes. Video frames with significant spatial and temporal information will draw more attention. Inspired by those researches, we proposed an RViT framework based on attention gate enabled RViT unit. The attention gate is designed to establish an interaction between spatial and temporal features and transfer temporal information through the hidden state.

In the following subsection, we will first discuss the

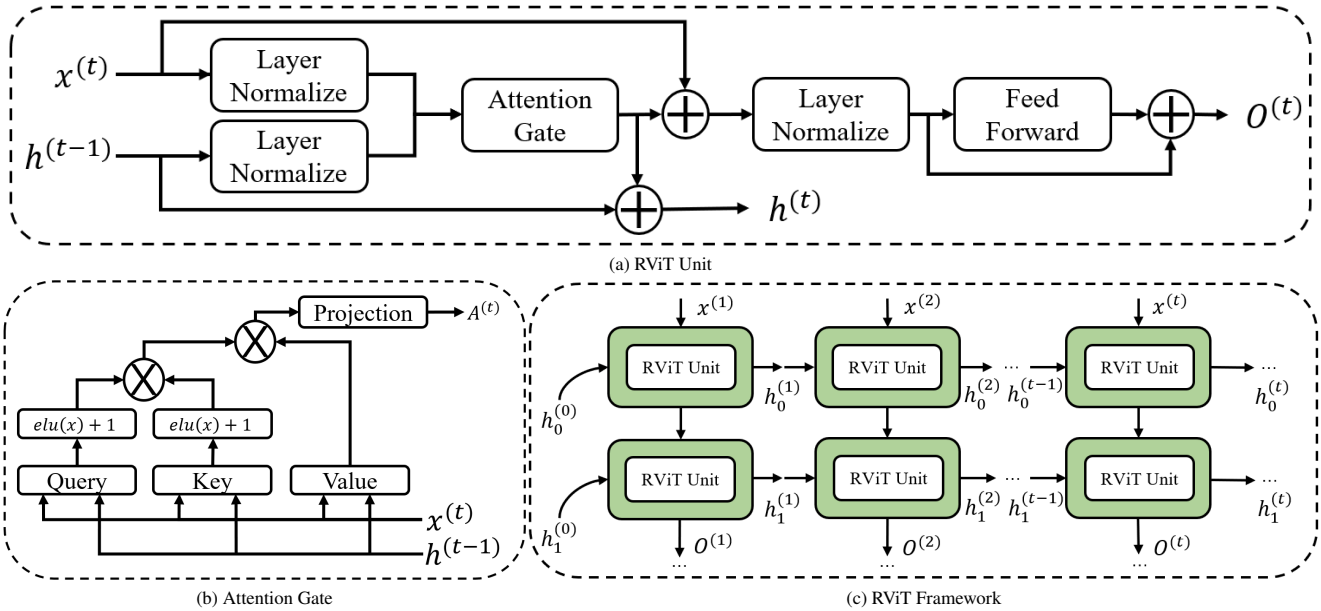


Figure 2. **RViT**. Figure (2a) shows a recurrent unit in our framework. Figure (2b) illustrates how attention gate process both current input $x^{(t)}$ and hidden state $h^{(t-1)}$. Figure (2c) illustrates an overview of the RViT framework, and gives a two layers RViT as an example, in where the spatial and temporal information is aggregated vertically and horizontally.

patch embedding of each frame in the pre-processing stage of RViT. The specifically designed attention gate then is introduced. The processing pipeline of the RViT unit is further discussed, followed by the token design. The whole framework will be presented finally.

3.1. Patch Embedding of Frames

In the pre-processing stage, the input image $X^{(t)} \in \mathbb{R}^{H\times W\times C}$ at the current frame will be decomposed into $P \times P$ non-overlapping patches and be flatten into vectors $x_p^{(t)} \in \mathbb{R}^{P^2\times D}$ and $D = \frac{H}{P} \cdot \frac{W}{P} \cdot C$.

Subsequently, an embedding layer is applied on the patched vector $x_p^{(t)}$, and followed by an appending of position encoding vector to generate the input vector $x^{(t)} \in \mathbb{R}^{P^2\times D}$ of the RViT unit:

$$x^{(t)} = \ell(x_p^{(t)}) + Pos_p \tag{1}$$

where ℓ is an embedding function (a convolution layer in our work), $Pos_p \in \mathbb{R}^{P^2\times D}$ is a learnable positional encoding vector, designed for spatial position encoding of each patch in a frame, and each frame shares same parameters of position encoding. Note that before input into the RViT unit, a token will be prepended to $x^{(t)}$ as discussed in section 3.4.

3.2. Attention Gate

Given the current frame $x^{(t)}$ and the hidden state $h^{(t-1)} \in \mathbb{R}^{P^2\times D}$ from the previous frame as the input, attention gate is designed to establish an interaction between them and generate the attended vector $a^{(t)} \in \mathbb{R}^{P^2\times D}$ as:

$$a^{(t)} = (\sigma(Q^{(t)}) + 1)(\sigma(K^{(t)})^T + 1)V^{(t)} \tag{2}$$

where $\sigma(\cdot)$ indicates the activation function $elu(\cdot)$, $Q^{(t)}, K^{(t)}, V^{(t)}$ are the Query/Key/Value matrices defined as:

$$\begin{aligned} Q^{(t)} &= x^{(t)}W_x^Q + h^{(t-1)}W_h^Q \\ K^{(t)} &= x^{(t)}W_x^K + h^{(t-1)}W_h^K \\ V^{(t)} &= x^{(t)}W_x^V + h^{(t-1)}W_h^V \end{aligned} \tag{3}$$

Multi-Head Attention [52] is also adopted in this work. We further extend Eq. 2 by concatenating q attention heads together:

$$A^{(t)} = \text{Concat}(a_1^{(t)}, \cdots, a_q^{(t)})W_{proj} \tag{4}$$

where $a_q^{(t)} = (\sigma(Q_q^{(t)}) + 1)(\sigma(K_q^{(t)})^T + 1)V_q^{(t)}$ and $a_q^{(t)} \in \mathbb{R}^{P^2\times \frac{D}{q}}$. A linear layer $W_{proj} \in \mathbb{R}^{D\times D}$ is adopted to project the attended vector. Note that we use linear attention instead of the SoftMax Attention to avoid gradient vanishing in this work. A diagram of the attention gate is shown in Figure 2b.

3.3. RViT Unit

An overview diagram of a single RViT unit is shown in Figure 2a. RViT unit consists of three steps. Firstly, the $x^{(t)}$

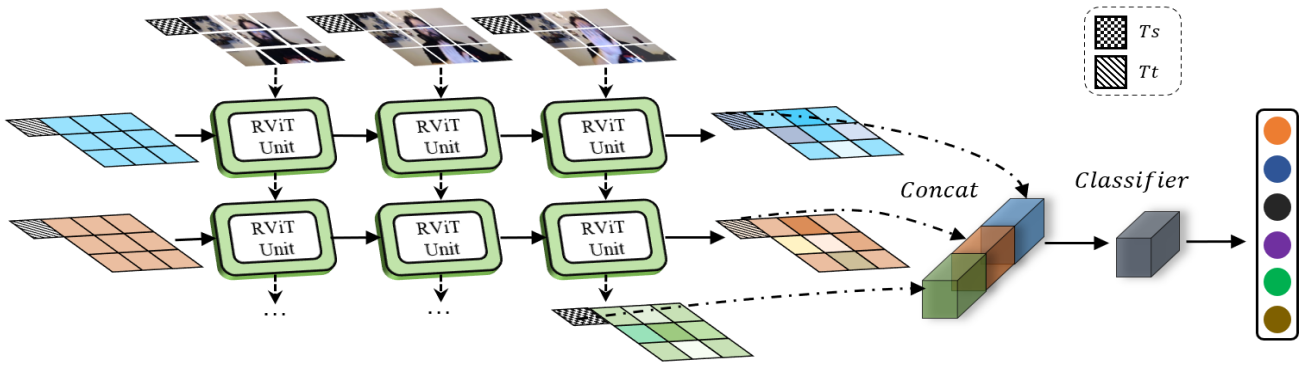


Figure 3. **The design of classification token**. This figure shows the complete schematic diagram of the `[class]` token transmission in spatial (vertical) and temporal (horizontal) directions. The output Tt token of each layer and the Ts token from the last moment will be concatenated for classification purposes.

and $h^{(t-1)}$ are passed by the Layer Normalization layer and passed to the attention gate. Next, the attention gate selectively preserves previous attention and append the new information from the current frame. Finally, The hidden state $h^{(t)} \in \mathbb{R}^{P^2 \times D}$ of the current frame then can be generated in a residual manner:

$$h^{(t)} = h^{(t-1)} + A^{(t)}, \tag{5}$$

And the output $O^{(t)} \in \mathbb{R}^{P^2 \times D}$ of current unit is produced by a Feed-Forward Network (FFN) with a residual connection, defined as:

$$O^{(t)} = f(o^{(t)}) + o^{(t)}, \tag{6}$$

where $o^{(t)}$ is the intermediate output defined as $o^{(t)} = x^{(t)} + A^{(t)}$, where $f(\cdot)$ denotes the FFN.

3.4. The Design of `[class]` Token

Similar to [11], we also incorporate an additional learnable token to serve as representation for classification purposes. Besides, the learnable token is also utilized to establish an interaction between spatial and temporal feature domains, as it is prepended to the input of the RViT unit. As shown in Figure 3, we use $Ts \in \mathbb{R}^D$ and $Tt \in \mathbb{R}^D$ to represent the learnable token in the spatial and temporal direction, respectively.

As shown in Figure 3, Ts tokens are prepended to each input frame $x^{(t)}$, and the Tt tokens are prepended to the initial hidden state in each layer before being feed into the RViT unit. Ts tokens are designed to aggregate features along the spatial dimension (vertically), while Tt tokens are utilized to aggregate features along the temporal dimension (horizontally).

The aggregated Tt tokens and the Ts from the last frame's output are concatenated together, and then a classification operation is performed on the concatenated tokens through a linear layer W_{class} (see Figure 3):

$$result = W_{class}(Concat(Tt_0^{(t)}, ..., Tt_n^{(t)}, Ts_n^{(t)})). \tag{7}$$

3.5. RViT Framework

Based on the above presented RViT unit and attention gate, a novel recurrent vision transformer can be formulated as Algorithm 1.

Algorithm 1 RViT: Recurrent Vision Transformer

Input:
$X^{(t)}$: The current input
$h_l^{(t-1)}$: The hidden state from previous unit in l layer
$h_l^{(0)}$: The initial hidden state in l layer
Ts,Tt : The learnable token for spatial and temporal
L, T : The total layers of RViT and input frames
ℓ, pos : Patch to embedding, Positional encoding
α : Attention gate
ln, f : Layer normalization, MLP

Output:
$h_l^{(t)}$ The state of t moment in l layer
$O_l^{(t)}$ The output of t momentin l layer

for $t = 1$ **to** T **do**
  $[x_1^{(t)} \cdots x_p^{(t)}] \leftarrow X^{(t)}$
  $x^{(t)} \leftarrow Concat\big(Ts, [\ell x_1^{(t)} \cdots \ell x_p^{(t)}]\big)$
  for $l = 1$ **to** L **do**
    $h_l^{(t-1)} \leftarrow Concat\big(Tt, h_l^{(0)}\big)$ **if** $t = 1$
    $x^{(t)} \leftarrow O_l^{(t)}$ **if** $l \neq 1$
    $A^{(t)} \leftarrow \alpha\big(ln(x^{(t)} + pos), ln(h_l^{(t-1)} + pos)\big)$
    $h_l^{(t)} \leftarrow A^{(t)} + h_l^{(t-1)}$
    $O_l^{(t)} \leftarrow f(ln(A^{(t)} + x^{(t)})) + (A^{(t)} + x^{(t)})$
  end for
end for

Note that we have some distinctive differences compared with other approaches: (i) Existing approaches generally process a batch of frames. For example, the 3D-ResNet and TimeSformer require a relatively long video sequence for

inference and training. While our method uses the recurrent unit to process the videos frame by frame; (ii) We incorporate the attention gate with hidden state into the RViT to aggregate the temporal attention recurrently without necessarily considering the video length. (iii) thanks to the frame flow processing, our method can work on both fixed-length and variant-length video clips without requiring large GPU memory.

4. Experiments and Results

4.1. Implement Details

Datasets To evaluate the proposed method, three public benchmark dataset for human action recognition task are adopted, including Kinetics-400(K400) [6, 30] ($\sim$240K training videos and $\sim$20K validation videos in 400 human action categories), Jester [39] ($\sim$120K training videos from 27 human gesture), Something-Something V2(SSv2) [20] ($\sim$168.9K training videos and 24.7K validation videos in 174 classes) and Charades [45](7985 videos for training and 1863 for testing in 157 classes).

Training For Kinetics-400, we first resize each video to 256×256, then sample a clip from the full-length video. Finally, a single clip is randomly cropped to 224×224 and randomly horizontally flipped. As our architecture is based on the vision transformer, we initialize the model with the ImageNet-21K pre-trained ViT model for Kinetics-400 experiments. The SSv2 dataset follows the same preprocessing pipeline as the above, except the pre-trained RViT model on Kinetics-400 is used. For the Jester dataset, the lengths of the videos might be insufficient to sample to 32 frames. Thus we pad short videos by randomly repeating the frame. We resize all frames to (112×112) without other transformations, and we train the model from scratch on the Jester dataset. Label smoothing and cross-entropy loss are adopted in training.

Top-1 and top-5 accuracy(%) are adopted for the evaluation on each validation dataset. The total model parameters, computation cost (Flops) and memory consumption for one-view inference are also included in the subsequent experiments. It should be noted that we use the official code [1, 7, 13, 14, 16, 38, 41, 50] (if available) when verifying other methods. Models with various settings are designed to verify the performance of our framework under different situations. The details of each config for different datasets are listed in Table 1.

Inference (i) For Kinetics-400 and Something-something V2, follow the pipeline from [15], we sample T random frames uniformly from a single video. Spatially, scales the shorter spatial side to 256 pixels and takes 3 crops of size 224×224 to cover the longer spatial axis. Temporally, we uniformly sample the long video into N clips and average the score from the last 1/3 frames when evaluating. The score of each test sample uses the average score of these $3 \times N$ predictions individually and takes the highest as the final prediction. In our work, we take each prediction as a single "view". (ii) For Jester dataset, we padded the short videos and randomly sampled long videos to the same length(T). Spatially, we resize each frame to 112×112 pixels without extra transformations. Take the highest prediction score evaluated from the last 10 frames as the final prediction. Note that for the inference time of RViT-XL ($64 \times 3 \times 3$) reported in Table 2, 3 temporal clips with 3 spatial crops (9 views in total) are used.

Model	Frame Size ($H \times W$)	Patch Size ($H \times W$)	Depth	Hidden	Head	Param (M)
RViT-S°	112×112	8×8	1	768	8	0.60
RViT°	112×112	8×8	2	768	8	1.15
RViT-L°	112×112	8×8	4	768	8	2.27
RViT	224×224	16×16	4	3072	12	36.8
RViT-L	224×224	16×16	8	3072	12	72.0
RViT-XL	224×224	16×16	12	3072	12	107.7

Table 1. **Model variants.** For the Jaster dataset, since the input frame size is 112×112, three types of models with 8×8 patch size are used (marked by ($\circ$)). For the K400 and SSv2 datasets, the frame size expands to 224×224.

4.2. Performance Evaluation

Kinetics-400 Performance results for Kinetics-400 are shown in Table 2. Compared to CNN-based methods and Transformer based methods, our methods can achieve state-of-the-art performance. Specifically:

- *Compare to CNN-based methods*, our best model (RViT-XL, 64×3×3) performs better ($1.7\% \sim 9.5\% \uparrow$) than CNN-based methods. The RViT-XL($32 \times 3 \times 1$) model achieves 80.3% of Top-1 accuracy while the flops are $3.49\times$ fewer than the SlowFast+NL. Meanwhile, our method outperforms the X3D-XL in Top-1 by 2.1%, with only $1.38\times$ larger in flops. In comparison with the CNN-based methods, our best model uses only 2.33GB memory ($2\times \sim 10\times$ less) in one-view inference.

- *Compare to Vision Transformer based methods*, our best model achieves **state-of-the-art** Top-1 accuracy (81.5%) compared to VIVIT(81.3%) and MViT-B(81.3%). Compared to ViViT, our model is $3\times$ lighter in terms of parameters with around 0.2% gain in Top-1 accuracy. We also outperform MViT-B by 0.2% in terms of top-1 accuracy at the cost of heavier parameters and Flops. In the aspect of memory cost, our model occupies remarkable less memory. Even our largest model use only 2.33GB ($3\times \sim 10\times$ less) during one-view inference. Our method requires extensive computation (11.96Tflops) because the inference is sampled from both spatial and temporal, which cause $2.49\times$ and $2.91\times$ more than ViViT and MViT-B, respectively.

Methods	Pre-Train	Top-1 (%)	Top-5 (%)	Param (M)	Flops (T)	Mem (G)
R(2+1)D* [51]	-	72.0	90.0	63.6	17.5	11.8
I3D* [6]	IN-1K	72.1	90.3	25.0	0.11	7.44
TSM [38]	IN-1K	74.1	N/A	24.3	0.65	5.98
S3D-G* [38]	-	74.7	93.4	N/A	N/A	6.75
NL I3D-101* [6]	IN-1K	77.7	93.3	25.0	0.36	7.73
ip-CSN-152* [50]	-	77.8	92.8	32.8	3.27	8.82
X3D-XL* [17]	-	79.1	93.9	11.0	1.45	>24
SlowFast+NL* [18]	-	79.8	93.9	59.9	7.02	4.25
TimeSformer* [3]	IN-21K	78.0	93.7	121.4	0.59	6.87
VTN* [40]	IN-21K	78.6	93.7	114.0	4.22	N/A
Mformer-B* [41]	IN-21K	79.7	94.2	114.0	11.0	7.3
MViT-B*, 32×3 [15]	-	80.2	94.4	36.6	0.85	10.7
En-VidTr-L* [60]	-	80.5	94.6	N/A	N/A	N/A
TimeSformer-L* [3]	IN-21K	80.7	94.7	121.4	7.14	>24
ViViT* [1]	IN-21K	81.3	94.7	310.8	4.79	>24
MViT-B*, 64×3 [15]	-	81.3	95.1	36.6	4.10	>24
RViT, 32×3×1	IN-21K	78.1	93.5	36.8	0.69	1.94
RViT-L, 32×3×1	IN-21K	78.9	93.6	72.0	1.34	2.12
RViT-XL, 32×3×1	IN-21K	80.3	94.4	107.7	2.01	2.33
RViT-XL, 64×3×3	IN-21K	**81.5**	95.0	107.7	11.9	**2.33**

Table 2. **Performance comparison on K400**. In this table, we categorize these methods into CNN based and ViT based. We report the inference cost with total Flops. We evaluation the gigabyte memory consumption in a single "view". Models need to process all frames at once are marked with $(*)$.

Something-something V2 Table 3 tabulates the performance of CNN-based methods, Vision-Transformer-based methods and our model on the SSv2 dataset. Our proposed RViT model can achieve a 65.3% Top-1 accuracy, outperforms all the CNN-based methods with lower computation cost ($0.93\times$ less than blVNet and $0.17\times$ less than TEA). Compared to MViT, our best model achieves 0.2% and 0.3% performance gain in Top-1 and Top-5 accuracy at the cost of $3\times$ heavier parameters and $9\times$ larger flops.

Methods	Pre-Train	Top-1 (%)	Top-5 (%)	Param (M)	Flops (T)	Mem (G)
SlowFast R50* [18]	K400	61.9	87.0	34.1	0.19	3.35
SlowFast R101* [18]	K400	63.1	87.6	53.3	0.32	4.20
TSM [38]	K400	63.3	88.2	42.9	0.19	5.98
MSNet* [33]	IN-21K	64.7	89.4	54.6	0.07	6.54
TEA* [37]	IN-21K	65.1	89.9	54.6	2.10	N/A
blVNet* [16]	-	65.1	90.3	54.6	0.12	5.92
TimeSformer-L* [3]	IN-21K	62.4	N/A	121.4	5.1	>24
VidTr-L* [60]	-	63.0	N/A	N/A	10.5	N/A
ViViT-L* [1]	-	65.4	89.8	310.8	N/A	>24
Mformer-B* [41]	IN-21K	66.5	90.1	114.0	1.10	7.3
MViT-B*, 32×3 [15]	K400	67.1	90.8	36.6	0.51	10.7
MViT-B*, 64×3 [15]	K400	67.7	90.9	36.6	1.36	>24
MViT-B-24*, 32×3 [15]	K600	68.7	90.9	36.6	1.36	>24
RViT, 32×3×1	K400	65.3	89.4	36.8	0.69	1.94
RViT-L, 32×3×1	K400	66.1	90.2	72.0	1.34	2.12
RViT-XL, 64×3×1	K400	**67.9**	**91.2**	107.7	3.99	**2.33**

Table 3. **Performance comparison on Something-Somthing-V2**.We evaluation the gigabyte memory consumption in a single "view".

Charades As videos in the Charades dataset have an average length of 30 seconds, we adopt Charade for long-sequence video action recognition evaluation. As shown in Table 4, RViT achieves the accuracy of 66.1% on Charades, outperforms MoViNet (63.2%) by a large margin. Results on Charades prove that RViT is also capable of long-sequence video action understanding. This is attributed to the clear stage boundary between adjacent actions in a video, e.g., the boundary between sitting and drinking from a cup. Since there is less dependence between adjacent actions, forgetting of sitting action will not impair the recognition of drinking. This also justifies the usage of aggregated temporal features over global-attention-based temporal features in RViT.

Methods	Pre-Train	mAP(%)	Param(M)	Flops(T)
NonLocal [54]	IN-1K+K400	37.5	54.3	16.3
STRG+NL [55]	IN-1K+K400	39.7	58.3	18.9
Timeception [26]	K400	41.1	N/A	N/A
LFB+NL [56]	K400	42.5	122	15.9
SlowFast R101+NL [18]	K400	42.5	59.9	7.02
X3D-XL [17]	K400	43.4	11.0	1.45
MViT-B, 64×3 [15]	K400	46.3	36.4	13.7
AssembleNet-101 [44]	K400	58.6	53.3	1.20
X3D-XL [17]	K600	47.1	11.0	1.45
MViT-B-24, 32×3 [15]	K600	47.7	53.0	7.08
MoViNet-A6 [31]	K600	63.2	31.4	0.31
RViT-L,N×3	K400	**64.3**	72.0	N×0.042
RViT-XL,N×3	K400	**66.1**	107.7	N×0.063

Table 4. **Performance comparison on Charades**. N denotes the length of the video clip. The Mean of N on Charades $\approx$ 30s.

Jester Table 5 shows the performance comparison against vanilla methods on the Jester dataset. As the results suggest, our best method can achieve 92.31% of Top-1 accuracy with **less parameters (*2.27M*) and computation consumption (*0.44*Gflops)**, while TimeSformer and the best CNN model are $89.94\%(2.37\%\downarrow)$ and $90.75\%(1.56\%\downarrow)$, with $46.6M$ and $4.8M$ in parameters, $1.568G$ and $1.346G$ in flops, respectively. Noted that all models are trained from scratch.

Methods ($32\times112\times112$)	Top-1 (%)	Top-5 (%)	Param (M)	Flops (G)	Mem (G)
ConvLSTM [58]	82.76	94.23	7.6	59.2	2.37
TSN [53]	83.90	99.60	10.7	16	N/A
MobileNet-Small† [24]	84.69	98.70	2.30	0.42	1.90
ResNet3D-10* [21]	88.81	99.01	14.4	18.2	1.96
R(2+1)D-RGB* [51]	89.08	98.76	63.6	16.9	1.93
MobileNet-Large† [24]	89.40	99.11	15.8	1.98	1.92
TimeSformer* [3]	89.94	99.52	4.8	43.1	13.1
ResNet3D-18* [21]	89.96	99.76	33.3	34.6	2.08
ResNet3D-50* [21]	90.75	99.52	46.6	50.2	2.59
SE-ResNet3D* [21,25]	90.64	99.84	48.7	52.3	2.68
RViT-S°,32×3×1	89.47	98.73	0.60	3.84	**1.70**
RViT°,32×3×1	91.26	99.17	1.15	7.04	1.74
RViT-L°,32×3×1	**92.31**	**99.87**	**2.27**	14.1	1.76

Table 5. **Performance comparison on Jester**. We evaluation the gigabyte memory consumption in a single "view". (†) indicates the MobileNet accompany with the LSTM unit.

Classification Token Type	Accuracy(%)
Only Temporal Domain	88.16
Only Spatial Domain	79.94
Both Spatial and temporal	**92.31**

Table 6. **The accuracy comparison for different methods in Jester Val**. In this experiment, training use stander RViT model with different measures.

4.3. Ablation Studies

We use the Jeste dataset for our extensive ablation study in this section. The purpose of the ablation study is to demonstrate the following hypotheses: (i) The designed classification token in RViT can represent spatial-temporal information. We compared different indicators and selected the best design through experiments to study the impact of the Ts token on the results. (ii) The introduction of linear attention can prevent the gradient vanishing/exploding* during training and improve the stability in convergence. (iii) The attention map on each video frame is visualized, and the attention changes across time dynamically are also demonstrated.

The Design of `[class]` Token As discussed in section 3.4, by default, Tt and Ts are designed in the spatial and temporal direction, respectively. They serve as the representation for classification purposes and establish an interaction between spatial and temporal feature domains. In the ablation study for the classification token, we adopt different designs of the classification token to explore the impact of different token designs in terms of accuracy, convergence speed and stability.

Specifically, three different token designs are explored: (i) the spatial-only token, which only uses Ts from the output of the last layer in the final moment as the basis of classification; (ii) the temporal-only token, which only uses the Tt token; (iii) Both Ts and Tt are used (default).

According to table 6, spatial-only token leads to the lowest accuracy, while the combination of spatial Ts and temporal Tt tokens achieve the highest accuracy. However, only using the Ts token from the temporal layer leads to around 4% lower performance than the combination one.

The gradient vanishing/exploding In this paper, we propose to recurrent the standard ViT to achieve a similar mechanism to the original recurrent neural network [12]. As the RNN design might easily produce gradient vanishing, several methods have been explored to avoid the gradient vanishing on RViT.

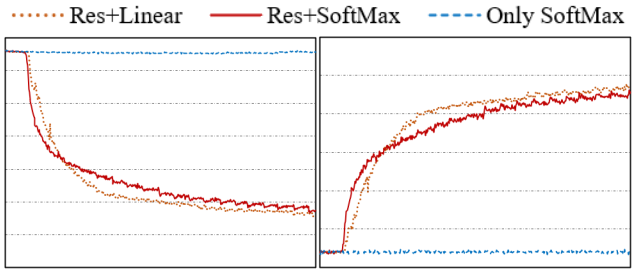
- ***Linear Attention***: The [29] uses the kernel-based self-attention and matrix product operation to calculate the self-attention weights. It reveals that gradient vanishing also exists in the transformer. In our case, the position of the action object in the adjacent frames might only change slightly. Thus the gradient may be aggregated in a fixed area position and lead to gradient vanishing/exploding. In addition, the characteristics of softmax in RNN will also lead to local gradient vanishing and high computational complexity. In order to avoid the gradient vanishing, we use linear attention to replace softmax attention as one of the remedies.

- ***Residual Connection*** *for Hidden State Transfer:* Inspired by relevant researches [10, 22], we add a residual connection between the the hidden state of each RViT unit. As shown in Figure 2a, the current hidden state $h^{(t)}$ from the current attention gate will be added with $h^{(t-1)}$ to form the final hidden state $h^{(t)}$.

*For simplicity, we only discuss gradient vanishing.

Residual Connection	Softmax Attention	Linear Attention	Acc(%)	Time(s)/ Epoch
-	✓	-	N/A	1386
-	-	✓	88.7	**1187**
✓	✓	-	87.9	1424
✓	-	✓	**91.3**	1191

Table 7. **Ablation study on residual connection and softmax/linear attention.** This table shows the evaluation result of model RViT° equipped with different components designed on the Jester-V1 dataset. The Time(s)/Epoch indicate the total training time of a single epoch. We implemented 10 epochs of training and presented the average time.



(a) Loss (b) Accuracy

Figure 4. **Loss and accuracy curve of variant settings.** This figure illustrates the impact of different components on training speed and accuracy.

The Table 7 shows models with different settings discussed above. As suggested by the result, the residual connection can prevent the gradient vanishing on both Softmax attention and linear attention. The introduction of linear attention can not only avoid the gradient vanishing but also reduce the computation and improve the accuracy (91.3%) significantly. According to Figure 4a, the SoftMax-only setting cannot achieve a converged training, while the introduction of residual connection together with the SoftMax attention can work well in RViT. The Figure 4b illustrates

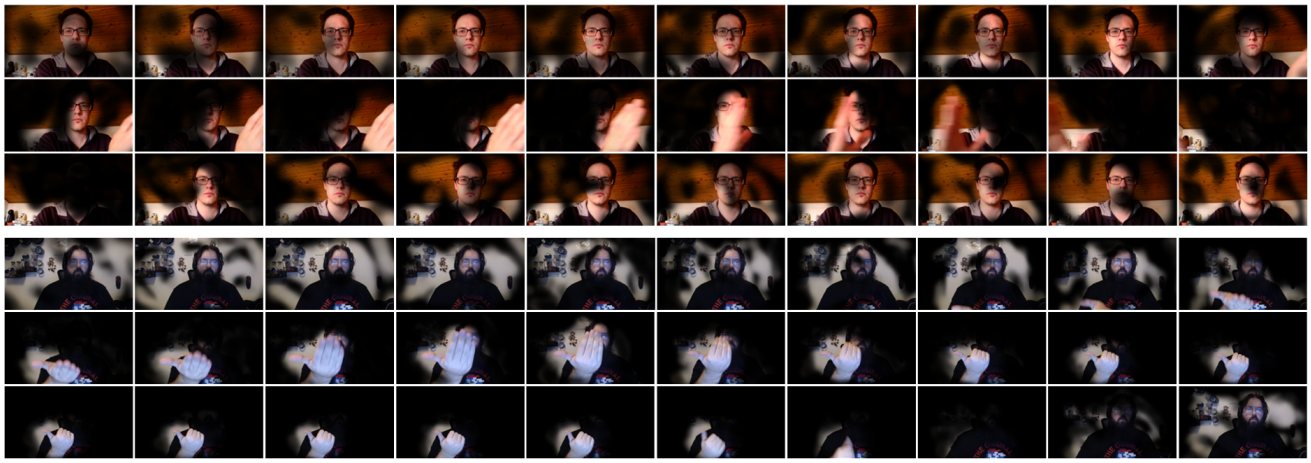


Figure 5. **Dynamic changes of attention**. The figure contains two video clip examples and intuitively shows the process of attention transmission in our framework. Each video contains 30 frames. The frames are displayed from left to right, top to bottom.

the accuracy performance under different settings. The results suggest that the residual connection with linear attention can achieve the best performance.

Dynamic changes in Attention We also explore to verify the effectiveness of our framework on modelling the spatial-temporal features of dynamic video. Here we use attention visualization of the transformer to verify the relationship of the attention maps among frames in the inference stage. Figure 5 and 6 show that the position of attention in the first few frames with slight motion is relatively scattered due to the poor classification confidence. As the gesture occurs in the following frames, attention is gradually focused on the hands' movements, driven by the aggravated attention information from previous frames.

5. Conclusion

In this work, considering the fact that existing approaches often process the videos in a clip-wise manner and fixed-length video clips are usually required, RViT is proposed for video understanding tasks. Specifically, the attention gate is incorporated into the RViT unit to transfer spatial information through the hidden state instead of extracting it from every frame. The video sequence is processed by executing the RViT unit in a recurrent manner. The proposed RViT can work in theory on both fixed-length and variant-length video clips properly without requiring large GPU memory.

We also evaluated our method on various public benchmark datasets. The results suggest that state-of-the-art performance can be achieved on video action recognition tasks with less GPU memory.

Though less GPU memory is needed, heavy computation may still be required to achieve considerable performance for our method, as the ViT structure is used as the basis of the RViT. On the other hand, the proposed RViT may experience information loss on the long video clips.

Adopting some classic RNN designs, such as LSTM and GRU, to RViT will become our future direction. Video prediction and video generation tasks will also be explored based on the proposed method.

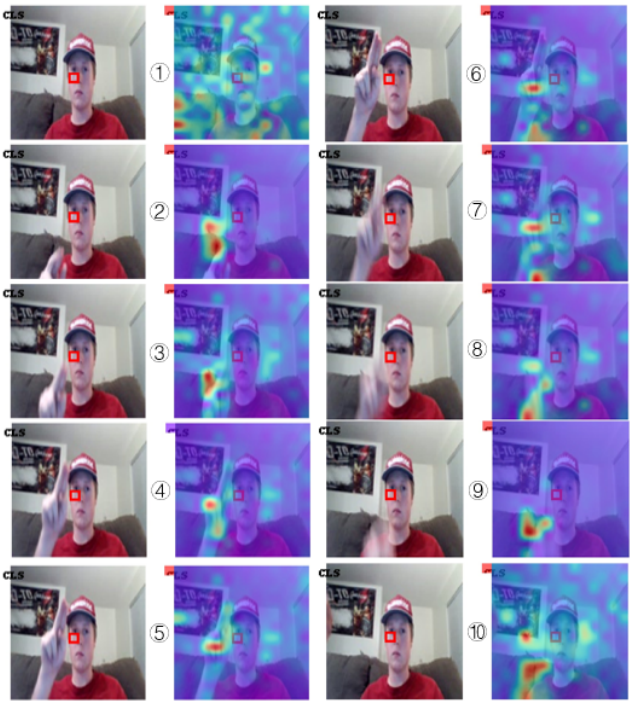


Figure 6. **Attention visualization**. The Figure shows the dynamic attention transmission in our network for moving objects. We visualize the attention map from each frame for transformer interpretability.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 2, 5, 6

[2] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. In *4th International Conference on Learning Representations, ICLR 2016*, 2015. 2

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021. 1, 2, 6

[4] Ali Borji, Dicky N Sihite, and Laurent Itti. What stands out in a scene? a study of human explicit saliency judgment. *Vision research*, 91:62–77, 2013. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 2, 5, 6

[7] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2, 2020. 5

[8] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *ICLR*, 2018. 2

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2018. 2

[10] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. *arXiv preprint arXiv:2103.03404*, 2021. 7

[11] 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. *ICLR*, 2021. 1, 2, 4

[12] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 7

[13] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. https://github.com/facebookresearch/slowfast, 2020. 5

[14] Haoqi Fan, Tullie Murrell, Heng Wang, Kalyan Vasudev Alwala, Yanghao Li, Yilei Li, Bo Xiong, Nikhila Ravi, Meng Li, Haichuan Yang, Jitendra Malik, Ross Girshick, Matt Feiszli, Aaron Adcock, Wan-Yen Lo, and Christoph Feichtenhofer. PyTorchVideo: A deep learning library for video understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. https://pytorchvideo.org/. 5

[15] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE international conference on computer vision*, 2021. 1, 2, 5, 6

[16] Quanfu Fan, Chun-Fu Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. In *Neural Information Processing Systems*, 2019. 2, 5, 6

[17] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. 1, 6

[18] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 1, 2, 6

[19] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016. 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 5842–5850, 2017. 5

[21] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 1, 2, 6

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 7

[23] HJ Heinze, George R Mangun, W Burchert, H Hinrichs, M Scholz, TF Münte, A Gös, M Scherg, S Johannes, H Hundeshagen, et al. Combined spatial and temporal imaging of brain activity during visual selective attention in humans. *Nature*, 372(6506):543–546, 1994. 1

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 6

[25] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 6

[26] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2019. 6

[27] Shuiwang Ji, Wei Xu, Ming Yang, and Kai . 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012. 1

[28] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 2

[29] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. 2, 7

[30] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5

[31] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021. 6

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 2

[33] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *European Conference on Computer Vision*, pages 345–362. Springer, 2020. 2, 6

[34] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*, 2019. 2

[35] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 2

[36] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 280–287, 2014. 2

[37] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2020. 6

[38] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 5, 6

[39] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 5

[40] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021. 1, 2, 6

[41] Mandela Patrick, Dylan Campbell, Yuki M Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, Jo Henriques, et al. Keeping your eye on the ball: Trajectory attention in video transformers. In *Neural Information Processing Systems*, 2021. 1, 2, 5, 6

[42] Michael I Posner and Steven E Petersen. The attention system of the human brain. *Annual review of neuroscience*, 13(1):25–42, 1990. 1

[43] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[44] Michael S Ryoo, AJ Piergiovanni, Mingxing Tan, and Anelia Angelova. Assemblenet: Searching for multi-stream neural connectivity in video architectures. *arXiv preprint arXiv:1905.13209*, 2019. 6

[45] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 5

[46] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014. 2

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[49] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 2

[50] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019. 5, 6

[51] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2, 6

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Infor-*

mation Processing Systems, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 2, 3

[53] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018. 6

[54] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2, 6

[55] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018. 6

[56] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*, 1(2):5, 2017. 6

[57] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 1

[58] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 2, 6

[59] Yun Zhai and Mubarak Shah. Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 815–824, 2006. 1, 2

[60] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13577–13587, 2021. 6