ActionCLIP: A New Paradigm for Video Action Recognition

Mengmeng Wang
Zhejiang University
mengmengwang@zju.edu.cn

Jiazheng Xing
Zhejiang University
jiazhengxing@zju.edu.cn

Yong Liu*
Zhejiang University
yongliu@iipc.zju.edu.cn

Abstract

The canonical approach to video action recognition dictates a neural model to do a classic and standard 1-of-N majority vote task. They are trained to predict a fixed set of predefined categories, limiting their transferable ability on new datasets with unseen concepts. In this paper, we provide a new perspective on action recognition by attaching importance to the semantic information of label texts rather than simply mapping them into numbers. Specifically, we model this task as a video-text matching problem within a multimodal learning framework, which strengthens the video representation with more semantic language supervision and enables our model to do zero-shot action recognition without any further labeled data or parameters requirements. Moreover, to handle the deficiency of label texts and make use of tremendous web data, we propose a new paradigm based on this multimodal learning framework for action recognition, which we dub "pre-train, prompt and fine-tune". This paradigm first learns powerful representations from pre-training on a large amount of web image-text or video-text data. Then it makes the action recognition task to act more like pre-training problems via prompt engineering. Finally, it end-to-end fine-tunes on target datasets to obtain strong performance. We give an instantiation of the new paradigm, ***ActionCLIP****, which not only has superior and flexible zero-shot/few-shot transfer ability but also reaches a top performance on general action recognition task, achieving 83.8% top-1 accuracy on Kinetics-400 with a ViT-B/16 as the backbone. Code is available at* `https://github.com/sallymmx/ActionCLIP.git`.

1. Introduction

Video action recognition is the first step of video understanding, and it is an active research area in recent years. We have observed that it mainly went through two stages, *feature engineering* and *architecture engineering*. Since there were no sufficient data for learning high-quality models before the birth of large datasets like Kinetics [5], early methods focused on the *feature engineering*, where researchers considered the temporal information inside the videos and used their knowledge to design specific hand-crafted representations [7, 42]. Then, with the advent of deep neural networks and large benchmarks, we are now in the second stage, *architecture engineering*. Lots of well-designed networks sprang up by reasonably absorbing the temporal dimension like two-stream networks [45], 3D convolutional neural networks (CNN) [12], compute-efficient networks [15] and transformer-based networks [2].

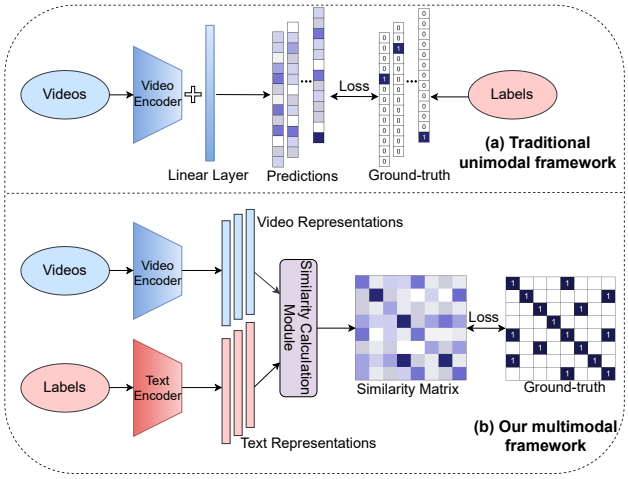

Figure 1. Existing unimodality pipeline (a) and our multimodal framework (b). They are different in the usage of labels. (a) maps labels into numbers or one-hot vectors while (b) exploits the semantic information of label text itself and tries to pull the corresponding video representation close to each other.

Though the features and network architectures have been well-studied in the last few years, they are trained to predict a fixed set of predefined categories within a unimodal framework as shown in Figure 1(a). This predetermined manner limits their generality and employment since additional labeled training data is required to transfer to any other new and unseen concepts. Instead of directly mapping labels to numbers like traditional works, learning from the

*Corresponding author.

raw text will be a promising solution which could be a much broader source of supervision and provide a more comprehensive representation. Reminiscent of how our humans do this job, we can recognize both known and unknown videos by associating the semantic information from the visual appearance to natural language sources rather than numbers. In this paper, we explore the natural language supervision in a multimodal framework as shown Figure 1(b) with two objectives, i) strengthening the representation of the traditional action recognition with more semantic language supervision, and ii) enabling our model to realize zero-shot transfer without any further labeled data or parameters requirements. Our multimodal framework includes two separate unimodal encoders for videos and labels and a similarity calculation module. The training objective is to pull the pairwise video and label representations close to each other, thus the learned representations will be more semantic than unimodal methods. In the inference phase, it becomes a video-text matching problem rather than a classical 1-of-N majority vote task and is capable of zero-shot prediction.

However, labels of existing fully-supervised action recognition datasets are always too succinct to construct rich sentences for language learning. Collecting and annotating new video datasets require huge storage resources and enormous human effort and time. On the other hand, a sea of videos with noisy but rich text labels are stored and generated on the web every day. Is there a way to energize the abundant web data for action recognition? Pre-training may be a solution that is demonstrated in ViViT [2]. But it is not easy to pre-train with a large magnitude of web data. It is expensive on storage hardware, computational resources and experiment cycles[1]. This triggers another motivation of this paper, could we directly adapt a pre-trained multimodal model into this task, avoiding the above dilemma? We find this is possible. Formally, we define a new paradigm "*pre-train, prompt, and fine-tune*" for video action recognition. Although it is appealing to *pre-train* the whole model end-to-end with large-scale video-text datasets such as HowTo100M [32], we are restricted by the enormous computation cost. Luckily, we find it is also worked to use a pre-trained model. Here we use the word "*pre-train*" rather than "*pre-trained*" in the new paradigm to keep the pre-training function. Then, instead of adapting the pre-trained model in specific benchmarks by substituting the final classification layers and objective functions, we reformulate our task to look more like those solved during the original pre-training procedure via *prompt*. Prompt-based learning [25] is regarded as a sea change to natural language processing (NLP), but it is not active in vision tasks, especially has not been exploited in action recognition. We believe it will have attractive prospects in many vision-text-related tasks and explore it in action recognition here. Finally, we *fine-tune* the whole model on target datasets. We implement an instantiation of this paradigm, **ActionCLIP**, which employs CLIP [36] as the pre-trained model. It obtains a top performance of 83.8% top-1 accuracy on Kinetics-400. Our contributions can be summarized as follows:

- We formulate the action recognition task as a multimodal learning problem rather than a traditional unimodal classification task. It strengthens the representations with more semantic language supervision and enlarges the generality and employment of the model in zero-shot/few-shot situations.
- We propose a new paradigm for action recognition, which we dub "*pre-train, prompt, and fine-tune*". In this paradigm, we could directly reuse powerful large-scale web data pre-trained models by designing appropriate prompts, significantly reducing the pre-training cost.
- Comprehensive experiments demonstrate the potential and effectiveness of our method, which consistently outperforms the state-of-the-art methods on several public benchmark datasets.

2. Related Works

2.1. Video Action Recognition

We have observed that video action recognition mainly went through two stages, *feature engineering* and *architecture engineering*. In the first stage, lots of hand-craft descriptors are designed for spatio-temporal representations, like Cuboids [7], 3D 3DHOG [17] and Dense Trajectories [42]. However, these features lack generalization since they are not end-to-end learned in large-scale datasets. Now we are in the second stage, *architecture engineering*. We coarsely classify these architectures into four categories, two-stream networks, 3D CNNs, compute-efficient networks and transformer-based networks. Two-stream-based methods [13, 45, 49] are introduced to model appearance and dynamics separately with two networks and fuse two streams through the middle or at last. 3D CNNs [5, 6, 12, 40] intuitively learn spatiotemporal features from RGB frames directly which extend the common 2D CNNs with an extra temporal dimension. Due to the heavy computational burden of 3D CNNs, many compute-efficient networks are designed to find the trade-off between precision and speed [41, 51, 55, 15, 19, 23]. Transformer-based networks [2, 3, 34, 37, 9] employ and modify recent strong vision transformers to jointly encode the spatial and temporal features. Yet, most works of both stages are unimodal, without considering the semantic information contained in the

[1] [8] reports that pre-training a ViT-H/14 model on JFT takes 2.5k TPUv3-core-days

labels. We propose a new paradigm "*pre-trained, prompt, and fine-tune*" based on a video-text multimodal learning framework for action recognition, which sheds light on the language modeling of label words.

2.2. Vision-text Multi-modality in Action Recognition

Vision-text multi-modality is a hot topic in several vision-text related fields recently, like pre-training [20, 21], vision-text retrieval [10, 31] and so on. Video action recognition could be interpreted as a text-insufficient video-to-text retrieval problem. Therefore, it may also be feasible to apply vision-text multi-modal learning in this task. However, to the best of our knowledge, we have not found mature and effective methods from this perspective in general video action recognition. We do find several vision-text multi-modality works in self-supervised video representation learning [30, 1] and zero-shot action recognition [35, 4, 53]. Yet, the former is prone to just learn a strong pre-trained video representation with a large web dataset and still neglects the label texts features when doing specific classification, just attaching and learning a linear classifier on top of the learned vision representation. The latter mainly concentrates on the embedding space designation with a pre-trained vision model and a simple text embedding like Word2Vec [33], paying less attention to the upstream general action recognition task. Different from them, in this paper, we focus on the vision-text multi-modality learning in general action recognition and build a bridge for it and zero-shot/few-shot action recognition.

3. Method

3.1. Multimodal Learning Framework

Previous comprehensive video action recognition methods treat this task as a classic and standard 1-of-N majority vote problem, mapping labels into numbers. This pipeline completely ignores the semantic information contained in the label texts. We instead model this task as a video-text multimodal learning problem, in contrast to pure video modeling. We believe learning from the supervision of natural language could not only enhance the representation power but also enable flexible zero-shot transfer.

Formally, given an input video $\boldsymbol{x}$ and a label $\boldsymbol{y}$ from a predefined label set $\mathcal{Y}$, the prior works usually train a model to predict the conditional probability $P(\boldsymbol{y}|\boldsymbol{x}, \theta)$ and turn $\boldsymbol{y}$ into a number or a one-hot vector to indicate its index of the whole label set length $|\mathcal{Y}|$. In the inference phase, the highest-scoring index of the prediction is regarded as the corresponding category. We try to break this routine and model the problem as $P(f(\boldsymbol{x}, \boldsymbol{y})|\theta)$, where $\boldsymbol{y}$ is the original words of the label and f is a similarity function. Then, the testing is more likely a matching process, the label words of the highest similarity score is the classification result:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}\in\mathcal{Y}}{argmax} P(f(\boldsymbol{x}, \boldsymbol{y})|\theta) \tag{1}$$

As shown Figure 1(b), we learn separate unimodal encoders g_V, g_W for video and label words inside a dual-stream framework. The video encoder g_V extracts spatio-temporal features for the visual modality and could be any well-designed architectures. The language encoder g_W is used to extract features of input label texts and could be a wide variety of language models. Then, to pull the pairwise video and label representations close to each other, we define symmetric similarities between the two modalities with cosine distances in the similarity calculation module:

$$s(\boldsymbol{x}, \boldsymbol{y}) = \frac{\mathbf{v} \cdot \mathbf{w}^\top}{\|\mathbf{v}\| \, \|\mathbf{w}\|}, s(\boldsymbol{y}, \boldsymbol{x}) = \frac{\mathbf{w} \cdot \mathbf{v}^\top}{\|\mathbf{w}\| \, \|\mathbf{v}\|} \tag{2}$$

where $\mathbf{v} = g_V(\boldsymbol{x})$ and $\mathbf{w} = g_W(\boldsymbol{y})$ are encoded features of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Then the softmax-normalized video-to-text and text-to-video similarity scores can be calculated as:

$$\begin{aligned} p_i^{\boldsymbol{x}2\boldsymbol{y}}(\boldsymbol{x}) &= \frac{\exp(s(\boldsymbol{x}, y_i)/\tau)}{\sum_{j=1}^{N} \exp(s(\boldsymbol{x}, y_j)/\tau)}, \\ p_i^{\boldsymbol{y}2\boldsymbol{x}}(\boldsymbol{y}) &= \frac{\exp(s(\boldsymbol{y}, x_i)/\tau)}{\sum_{j=1}^{N} \exp(s(\boldsymbol{y}, x_j)/\tau)} \end{aligned} \tag{3}$$

where τ is a learnable temperature parameter and N is the number of training pairs. Let $\boldsymbol{q}^{\boldsymbol{x}2\boldsymbol{y}}(\boldsymbol{x}), \boldsymbol{q}^{\boldsymbol{y}2\boldsymbol{x}}(\boldsymbol{y})$ indicate the ground-truth similarity scores, where the negative pair has a probability of 0 and the positive pair has a probability of 1. Since the amount of videos are much larger than the fixed labels, it will inevitably appear multiple videos belonging to one label in a batch. Therefore, it may exist more than one positive pair in both $q_i^{\boldsymbol{x}2\boldsymbol{y}}(\boldsymbol{x})$ and $q_i^{\boldsymbol{y}2\boldsymbol{x}}(\boldsymbol{y})$. It is not proper to regard the similarity score learning as a 1-in-N classification problem with cross-entropy loss. Instead, we define the Kullback–Leibler (KL) divergence as the video-text contrastive loss to optimize our framework as:

$$\mathcal{L} = \frac{1}{2}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[\mathrm{KL}(\boldsymbol{p}^{\boldsymbol{x}2\boldsymbol{y}}(\boldsymbol{x}), \boldsymbol{q}^{\boldsymbol{x}2\boldsymbol{y}}(\boldsymbol{x})) + \mathrm{KL}(\boldsymbol{p}^{\boldsymbol{y}2\boldsymbol{x}}(\boldsymbol{y}), \boldsymbol{q}^{\boldsymbol{y}2\boldsymbol{x}}(\boldsymbol{y}))] \tag{4}$$

where $\mathcal{D}$ is the whole training set. Based on the multimodal framework, we can simply carry out zero-shot prediction as the normal testing process in Equation 1.

3.2. The New Paradigm

When considering the above multimodal learning framework, we need to consider the deficiency of label words. The most intuitive way is to take advantage of vast web image-text or video-text data. To cater for this, we propose a new "*pre-train, prompt and fine-tune*" paradigm for action recognition.

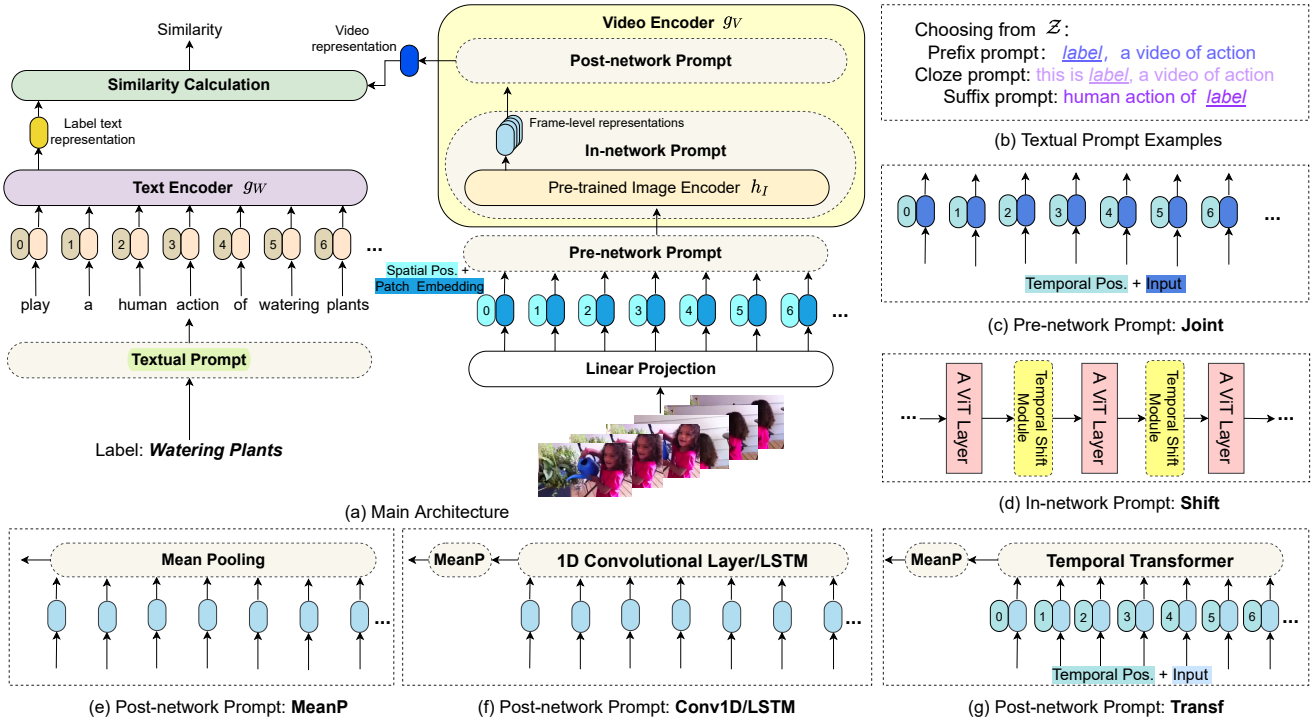

Figure 2. Overview of **ActionCLIP**. We present the whole architecture in (a), which consists of two single-modal encoders, a similarity calculation module and all possible prompt locations. (b) shows several examples of the textual prompt. (c) and (d) are the in-network and pre-network visual prompts details, respectively. (e),(f) and (g) give the details of post-network visual prompts, *MeanP*, *Conv1D*, *LSTM* and *Transf. Pos.* is short for positional.

Pre-train. As prior arts suggested, pre-training has a large impact on vision-language multimodal learning [28, 20, 21, 16]. Since the training data is directly collected from the web, one of the hot topics is to design appropriate objectives to handle these noisy data during this procedure. We find there are mainly three upstream pre-training proxy tasks in the pre-training procedure: multimodal matching (MM), multimodal contrastive learning (MCL) and masked language modeling (MLM). MM predicts whether a pair of modalities is matched or not. MCL aims to draw pairwise unimodal representations close to each other. MLM utilizes the features of both modalities to predict the masked words. However, this paper does not focus on this step due to the restriction of enormous computation cost. We directly choose to apply a pre-trained model and make efforts on the following two steps.

Prompt. Prompt in NLP means the original input is modified using a template into a textual string prompt that has some unfilled slots to fill with expected results. Here we borrow the word "*prompt*" for the meaning of *adjusting and reformulating the downstream tasks to act more like the upstream pre-training tasks*. Notably, the traditional practice is adapting the pre-trained model to the downstream classification task via attaching a new linear layer to the pre-trained feature extractor, which is reversed to ours. Here we make two kinds of prompts, textual prompt and visual prompt. The former is significant for label text extension. Given a label $\boldsymbol{y}$, we first define a set of permissible values $\mathcal{Z}$, then the prompted textual input $\boldsymbol{y}'$ is obtained by a filling function $f_{fill}(\boldsymbol{y}, \boldsymbol{z})$, where $\boldsymbol{z} \in \mathcal{Z}$. There are three varieties of f_{fill}, *prefix prompt*, *cloze prompt* and *suffix prompt*. They are classified based on the filling locations. For visual prompt, its designation mainly depends on the pre-trained model. If the model is pre-trained on video-text data, it is almost no extra reformulation for the visual part since the model is already trained to output video representations. While if the model is pre-trained with image-text data, then we should empower the model to learn the important temporal relationship of videos. Formally, given a video $\boldsymbol{x}$, we introduce prompt function as $f_{tem}(h_I(x))$, where h_I is the visual encoding network of pre-trained models. Similarly, f_{tem} has three variants based on where it works against h_I, *pre-network prompt*, *in-network prompt* and *post-network prompt*. With the elaborate designation of *prompt*, we could even avoid the above unreachable computational "*pre-train*" step by keeping the learned ability of a pre-trained model. Note that in the new paradigm, the pre-trained model should not be largely modified due to *catas-*

trophic forgetting [29], where the pre-trained model loses its ability to do things that it was able to do in the pre-training. We also demonstrate this point in our experiments.

Fine-tune. When there are sufficient downstream training datasets like Kinetics, it is no doubt that fine-tuning on specific datasets will dramatically improve the performance. Also, if the prompt introduces extra parameters, it is necessary to train these parameters and learn with the whole framework end-to-end.

3.3. New Paradigm Instantiation Details

Each component of the new paradigm has a wide variety of choices. As presented in Figure 2, we show an instantiation example here and conduct all the experiments with this instantiation.

We employ a firsthand pre-trained model, CLIP [36] to avoid the enormous computational resources at the first pre-training step. This instantiation model is called **ActionCLIP** as shown in Figure 2(a). CLIP is an efficient image-text representation trained with the MCL task, similar to our multimodal learning framework. Figure 2(b) shows concrete examples of the textual prompts used in the instantiation. We define $\mathcal{Z}$ to be K discrete manual sentences which is the most natural way based on human introspection. Then the prompted input $\boldsymbol{y}'$ is fed into the language encoder g_W that is the same with pre-trained language model h_W. For the vision model, based on the pre-trained image encoder h_I of CLIP, we employ three types of visual prompts as follows.

Pre-network Prompt. This type operates on the inputs before feeding into the encoder, as shown in Figure 2(c). Given a video $\boldsymbol{x}$, we simply forward all spatio-temporal tokens extracted from the video through the visual encoder to jointly learn spatio-temporal attentions. Except for the spatial positional embedding, an extra learnable temporal positional embedding will be added to the token embedding to indicate the frame index. g_V could use the original pre-trained image encoder h_I. We call this type *Joint* for short.

In-network Prompt. We attempt a parameter-free prompt abbreviated as *Shift* for this type as shown in Figure 2(d). We introduce the temporal shift module [24], which shifts part of the feature channels along the temporal dimension and facilitates information exchanged among neighboring input frames. We insert the module between every two adjacent layers of g_V. The architecture and pre-trained weights of g_V could directly reuse h_I since this module brings no parameters.

Post-network prompt. Given a video $\boldsymbol{x}$ with F extracted frames, we sequentially encode spatial and temporal features with two separate encoders in this prompt. The first is a spatial encoder g_V^{sp} which is responsible for only modeling interactions between tokens extracted from the same temporal index. We use h_I as our g_V^{sp}. The extracted frame-level representations $u_i \in \mathbb{R}^d$ are then concatenated into $\mathbf{u} \in \mathbb{R}^{F\times d}$, and then fed to a temporal encoder g_V^{te} to model interactions between tokens from different temporal indices. We offer four choices for g_V^{te}, *MeanP*, *Conv1D*, *LSTM* and *Transf*, presented in Figure 2(e-g). *MeanP* is short for mean pooling on the temporal dimension. *Conv1D* is a 1d convolutional layer applied on the temporal dimension. *LSTM* is a recurrent neural network and *Transf* means a L_t layer temporal vision transformer encoder. Since the temporal dimensions of *Conv1D*, *LSTM* and *Transf* keep the same with input $\mathbf{u}$, the subsequent operations are the same as *MeanP*.

Then we end-to-end fine-tune the whole network with the training objective Equation 4.

4. Experiments

4.1. Experimental Setup

Network architectures. Our textual encoder g_W follows that of CLIP which is a 12-layer, 512-wide Transformer with 8 attention heads and the activations from the highest layer at [EOS] are treated as the feature representation $\mathbf{w}$. We use ViT-B/32 and ViT-B/16 of CLIP's visual encoder h_I. They are all 12-layer vision transformers, with different input patch sizes of 32 and 16 respectively. The [Class] token of their highest layers' outputs are used. We use K=18 permissible values $\mathcal{Z}$ for textual prompt. For visual prompts, the layer of *Conv1D* and *LSTM* is 1, *Transf* has L_t=6 layers. Two versions of *Transf* are implemented, they are different in not using or using [Class] token. We distinguish them as *Transf* and *Transf*$_{cls}$.

Training. We use AdamW optimizer with a base learning rate of 5×10^{-6} for pre-trained parameters and 5×10^{-5} for new modules with learnable parameters. Models are trained with 50 epochs and the weight decay is 0.2. The learning rate is warmed up for 10% of the total training epochs and decayed to zero following a cosine schedule for the rest of the training. The spatial resolution of the input frames is 224×224. We use the same segment-based input frame sampling strategy as [46] with 8, 16 or 32 frames. Even the largest model of our method, ViT-B/16 could be trained with 4 NVIDIA GeForce RTX 3090 GPUs on Kinetics-400 when inputting 8 frames, and the training process takes about 2.5 days. Compared to X3D and SlowFast, both trained with 128 GPUs for 256 epochs, our training is much faster and requires fewer GPUs ($\sim$30).

Inference. The input resolution is 224×224 in all the experiments. Following [15], we use multi-view inference with 3 spatial crops and 10 temporal clips of each video only for the best performance model. The final prediction is from the averaged similarity scores of all views.

4.2. Ablation Experiments

In this section, we do extensive ablation experiments to demonstrate our method with the instantiation, **ActionCLIP**. Models in this section use 8-frame input, Transf for temporal modeling, ViT-B/32 as the backbone and single view testing on Kinetics-400, unless specified otherwise.

Is the "*multimodal framework*" helpful? To compare with the traditional video-unimodal 1-of-N classification model, we implement a variant called *unimodality* which has the same backbone, pre-trained weights and temporal modeling strategy (before the final linear layer) with our **ActionCLIP**. The results are shown in Table 1. When exploiting the semantic information of label texts with our multimodal learning framework, it dramatically improves the performance with 2.91% top-1 accuracy gains, demonstrating that the multimodal framework is helpful to learn powerful representations for action recognition.

Table 1. Ablation of the "*multimodal framework*".

	Unimodality	**ActionCLIP**
Top-1	75.45	78.36
Top-5	92.51	94.25

Is the "*pre-train*" step important? In Table 2, we validate the impact of this step by experimenting with random initialized or CLIP pre-trained vision and language encoders. In particular, from the large gap (40.10% *vs.* 78.36%) between model V3 and V4, we find that the visual encoder needs proper initialization, otherwise the model will fail to obtain a strong performance. The language encoder has a smaller influence, since model V2 could also get a comparative result (76.63%) compared with model V4 (78.36%). When both the visual and language encoders are randomly initialized, model V1 is hard to learn a good representation and drops a large margin of 41.4% from model V4. Therefore, the final conclusion is that the "*pre-train*" step is important, especially for the visual encoder.

Table 2. Ablation of the "*pre-train*" step.

Model	Language	Vision	Top1	Top5
V1	random	random	36.96	63.02
V2	random	CLIP	76.63	91.94
V3	CLIP	random	40.10	66.35
V4	CLIP	CLIP	78.36	94.25

Table 3. Ablation of the textual *prompt*.

	only label	textual prompt
Top-1	77.82	78.36
Top-5	93.95	94.25

Is the "*prompt*" step important? Table 3 shows the results of textual prompt. It can be seen that using only the label words drops 0.54% compared with using textual prompt, demonstrating the validness of this simple, discrete and human-comprehensible textual prompt. For the visual prompt, note that *MeanP* is the simplest temporal fusion way and we compare other visual prompts with it. As shown in Table 4, we find *Joint* and *Shift* obviously decrease the performance by 2.74% and 5.38%, respectively. We believe the reason is the *catastrophic forgetting* phenomenon since the input pattern is changed in *Joint* and the features of pre-trained image encoder h_I are changed in *Shift*. These operations may break the original learned strong representations and yield performance drop. Post-network prompts are more suitable and safer options to keep the learned character. Specifically, *LSTM* and *Conv1D* cause a negligible top-1 drop but they all improve the top-5 accuracy. *Transf*$_{cls}$ and *Transf* improve the top-1 results with 1.01% and 1.25%. We choose *Transf* as our final visual prompt since it has the best results. In a word, the designation of *prompt* is significant since proper prompts could avoid *catastrophic forgetting* and maintain the representation power of existing pre-trained models, giving a shortcut to the usage of tremendous web data.

Table 4. Ablation of the visual *prompt*.

Visual prompt		Top-1	Top-5
Pre-network	Joint	74.37	93.09
In-network	Shift	71.73	91.07
Post-network	MeanP	77.11	93.79
	LSTM	77.09	93.86
	Conv1D	77.04	94.06
	Transf$_{cls}$	78.12	94.25
	Transf	78.36	94.25

Is the "*fine-tune*" step important? We demonstrate this step by separately freezing the parameters of the pre-trained language encoder h_W and image encoder h_I. The results are presents in Table 5. When the two encoders are all frozen and only the visual prompt *Transf* is trained, the performance decreases 6.15% on top-1 accuracy. When all the parameters are end-to-end fine-tuned, we obtain the best results of 78.36%. It will have a negative influence on the accuracy if either of the pre-trained encoders is frozen. Therefore, the "*fine-tune*" step is indeed crucial to specific datasets, which is consistent with our perceptual intuition.

Backbones and input frames. In Table 6, we experiment

Table 5. Is the "*fine-tune*" step important? "✓" means do fine-tuning while "×" means fixing the parameters without fine-tuning.

Model	Language	Video	Top-1	Top-5
V1	×	×	72.21	91.61
V2	×	✓	73.88	91.61
V3	✓	×	73.78	92.17
V4	✓	✓	78.36	94.25

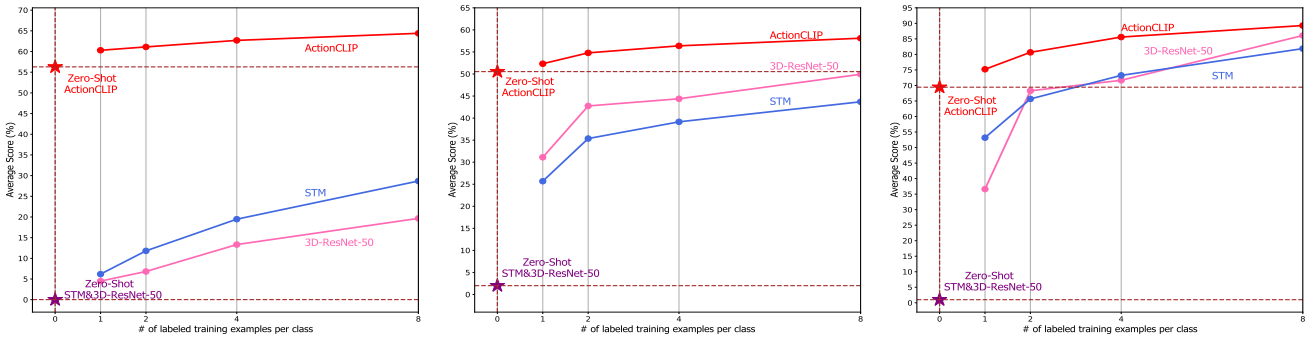

Figure 3. Zero-shot/few-shot results on Kinetics-400 (left), HMDB-51 (middle) and UCF-101 (right). **ActionCLIP** leads the performance under these hard data-poor circumstances. It can do zero-shot recognition on all three datasets while STM and 3D-ResNet-50 are not available under this condition. Also, **ActionCLIP** is good at few-shot classification where the performance gap is obvious compared to the other two methods. Brown dashed lines demonstrate zero-shot accuracy.

ActionCLIP with different backbones and input frames configurations. The input frames vary from 8, 16 to 32. Two different backbones are used, ViT-B/32 and ViT-B/16. The conclusion is intuitive that larger models and more input frames yield better performance.

Table 6. Influence of backbones and input frames.

Backbone	Input frames	Top1	Top5
ResNet-50	8	73.31	92.02
ViT-B/32	8	78.36	94.25
ViT-B/16	8	81.09	95.49
	16	81.68	95.87
	32	82.32	96.20

4.3. Runtime Analysis

For different backbones and input frame configurations, we present their model sizes, FLOPs and inference speeds in Table 7. The textual encoder of all backbones has the same architecture, which has 37.8M parameters. We show the whole parameter in the table. Notably, ViT-B/32 has a little more parameters than ViT-B/16, which comes from the linear projection layer before feeding into the vision transformer. While ViT-B/32 has much a faster inference speed (3.3×) and fewer FLOPs (4×) than ViT-B/16. Moreover, we provide two very recent methods for comparison, TimeSformer [3] and ViViT [2] with their highest configurations which obtain similar accuracy with **ActionCLIP**. Specifically, compared with the highest configuration of **ActionCLIP**, TimeSformer needs more input frames (3×) and much more computational FLOPs (12.7×) to obtain its best performance, which is still worse than **ActionCLIP** (82.3% *vs.* 80.7%). Similarly, ViViT has much more computational FLOPs (7.1×) to obtain its best results with a larger input solution 320×320, while **ActionCLIP**'s input is 224×224 and it surpasses ViViT with 1% top-1 accuracy gap and runs faster (3.1×) than ViViT. In conclusion, **ActionCLIP** is a cost-effective and efficient method for action recognition.

Table 7. Parameters, FLOPs and inference speed comparison. All the results of **ActionCLIP** are tested on one NVIDIA GeForce RTX 3090 GPU and use single-view testing.

Backbone	Frames	Top-1	GFLOPs	Params	Runtime
TimesSformer-L	96	80.7	7140	-	-
ViViT-L/16 320	32	81.3	3992	-	4.2V/s
ViT-B/32	8	78.4	35.4	144.1M	144.7V/s
ViT-B/16	8	81.1	140.8	141.7M	43.2V/s
	16	81.7	281.6	141.7M	21.2V/s
	32	**82.3**	563.1	141.7M	13.0V/s

4.4. Zero-shot/few-shot Recognition

In this section, we demonstrate the attractive zero-shot/few-shot recognition ability of our **ActionCLIP** (ViT-B/16). We implement two representative methods for comparison, STM [15] which is a well-designed temporal-encoded 2D network, and 3D-ResNet-50 which is the slow path of SlowFast [12]. We use 8-frame input, single view inference in all models of this section. We first conduct the zero-shot/few-shot experiments on Kinetics-400. **ActionCLIP** uses pre-trained model of CLIP with *MeanP* visual prompt (since *Transf* has no pre-trained parameters), STM and 3D-ResNet-50 are pre-trained on ImageNet. Then, we validate on UCF-101 and HMDB-51 with Kinetics-400 pre-trained models (**ActionCLIP** uses *Transf* here). As shown in Figure 3, the results demonstrate the strong transfer power of **ActionCLIP** under these data-poor conditions, while the traditional unimodality methods are not able to do zero-shot recognition and their few-shot performance is ineffective compared with **ActionCLIP** even pre-trained on large Kinetics-400.

Table 8. Comparison with previous work on Kinetics-400. "-" indicates the numbers are not available for us. "+" in the second column means the different input of two paths of the method.

Methods	Frames	Top-1	Top-5
I3D NL [47]	32	77.7	93.3
S3D-G [51]	64	74.7	93.4
SlowFast [12]	16+64	79.8	93.9
X3D-XXL [11]	16	80.4	94.7
TPN [52]	32	78.9	93.9
SmallBigNet [22]	32	77.4	93.3
CorrNet [43]	32	79.2	-
R(2+1)D [41]	32+32	75.4	91.9
TSM [24]	16	74.7	-
TEA [23]	16	76.1	92.5
STFT [19]	64	75.0	91.1
STM [15]	16	73.7	91.6
TANet [27]	16	79.3	94.1
TEINet [26]	16	76.2	92.5
TDN [44]	8+16	79.4	93.9
ViT-B-VTN [34]	250	79.8	94.2
MViT-B [9]	64	81.2	95.1
STAM [37]	64	80.5	-
TimeSformer-L [3]	96	80.7	94.7
ViViT-L/16x2 [2]	32	80.6	94.7
ViViT-L/16x2 (JFT)	32	82.8	95.3
ActionCLIP (ViT-B/16)	16	82.6	96.2
	32	**83.8**	**97.1**

Table 9. Comparison with previous work on Charades. "-" indicates the numbers are not available for us. "+" in the second column means the different input of two paths of the method.

Method	Frames	mAP
MultiScale TRN [54]	-	25.2
STM [15]	16	35.3
Nonlocal [47]	-	37.5
STGR+NL [48]	-	39.7
SlowFast 50 [12]	8+32	38.0
SlowFast 101+NL	16+64	42.5
X3D-XL (312) [11]	16	43.4
LFB+NL [50]	32	42.5
Timeception[14]	-	41.1
ActionCLIP (ViT-B/16)	32	**44.3**

4.5. Comparison with State-of-the-art Methods

In this section, we evaluate the performance of our method on a diverse set of action recognition datasets: Kinetics-400 [5], Charades [38], UCF-101 [39] and HMDB-51 [18]. ViT-B/16, Transf prompt and multi-view testing are used in **ActionCLIP**. The results of UCF-101 and HMDB-51 are shown in Appendix. **Kinetics-400.** Table 8 compares to prior methods on Kinetics-400. There are four parts in this table, corresponding to 3D-CNN-based methods, 2D-CNN-based methods, transformer-based methods and our method. According to the table, the third section achieves better results with strong vision transformers than the first and second parts. Among them, the first four methods build on the 12-layer ViT-B/16 model for parameter-accuracy balance, so do our **ActionCLIP**. ViViT instead uses a larger model, 24-layer ViT-L for better results. Also, it introduces JFT for pre-training for further gain. Our **ActionCLIP** achieves 82.6% top-1 accuracy with only 16-frame input, which exceeds all the methods in the first and second parts of the table and most transformer-based methods that may use more input frames like 250 frames of ViT-B-VTN. An interesting discovery is that our top-5 accuracy is always higher than other methods. We think this benefits from our multimodal framework's different inference process, which calculates the similarity between the semantic representations of videos and all labels. **ActionCLIP** further reaches a leading performance of 83.8% when increasing the input to 32 frames. We believe that more input frames, larger models and larger input resolutions will yield better results and leave it to future work. The current performance of **ActionCLIP** could already reveal the potential of the multimodal learning framework and the proposed new paradigm for action recognition.

Charades. This is a dataset with longer-range activities and it has multiple actions inside every video. We show Kinetics-400 pre-trained models in Table 9. Mean Average Precision (mAP) is used for evaluation. **ActionCLIP** achieves the top performance of 44.3 mAP, which demonstrates its effectiveness on multi-label video classification.

5. Conclusion

This paper provides a new perspective for action recognition by regarding it as a video-text multimodal learning problem. Unlike the canonical approaches that model the task as a video unimodality classification problem, we propose a multimodal learning framework to exploit the semantic information of label texts. Then, we formulate a new paradigm, i.e., "*pre-train, prompt, and fine-tune*" to enable our framework to directly reuse powerful large-scale web data pre-trained models, greatly reducing the pre-training cost. We implement an instantiation of the new paradigm, **ActionCLIP**, which has a superior performance on both general and zero-shot/few-shot action recognition. We hope our work could provide a new perspective for this task, especially raising attention on language modeling.

6. Acknowledgements

We would like to thank Zeyi Huang for his constructive suggestions and comments on this work.

References

[1] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelovic, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *NeurIPS*, 2(6):7, 2020.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

[4] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4613–4623, 2020.

[5] 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.

[6] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018.

[7] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021.

[10] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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*, pages 2000–2009, 2019.

[16] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv preprint arXiv:2102.03334*, 2021.

[17] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008.

[18] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[19] Sudhakar Kumawat, Manisha Verma, Yuta Nakashima, and Shanmuganathan Raman. Depthwise spatio-temporal stft convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[20] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021.

[21] Junnan Li, Ramprasaath R Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *arXiv preprint arXiv:2107.07651*, 2021.

[22] Xianhang Li, Yali Wang, Zhipeng Zhou, and Yu Qiao. Smallbignet: Integrating core and contextual views for video classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1092–1101, 2020.

[23] 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.

[24] 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.

[25] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[26] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recog-

nition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11669–11676, 2020.

[27] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. *arXiv preprint arXiv:2005.06803*, 2020.

[28] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

[29] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[30] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020.

[31] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018.

[32] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019.

[33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

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

[35] AJ Piergiovanni and Michael Ryoo. Learning multimodal representations for unseen activities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 517–526, 2020.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[37] Gilad Sharir, Asaf Noy, and Lihi Zelnik-Manor. An image is worth 16x16 words, what is a video worth? *arXiv preprint arXiv:2103.13915*, 2021.

[38] 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.

[39] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[40] Jonathan Stroud, David Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 625–634, 2020.

[41] 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.

[42] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[43] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 352–361, 2020.

[44] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1895–1904, 2021.

[45] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[46] 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.

[47] 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.

[48] 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.

[49] Yunbo Wang, Mingsheng Long, Jianmin Wang, and Philip S Yu. Spatiotemporal pyramid network for video action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1529–1538, 2017.

[50] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[51] 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.

[52] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020.

[53] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *Proceedings*

of the European Conference on Computer Vision (ECCV), pages 374–390, 2018.

[54] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.

[55] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Mict: Mixed 3d/2d convolutional tube for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 449–458, 2018.