Condensing a Sequence to One Informative Frame for Video Recognition

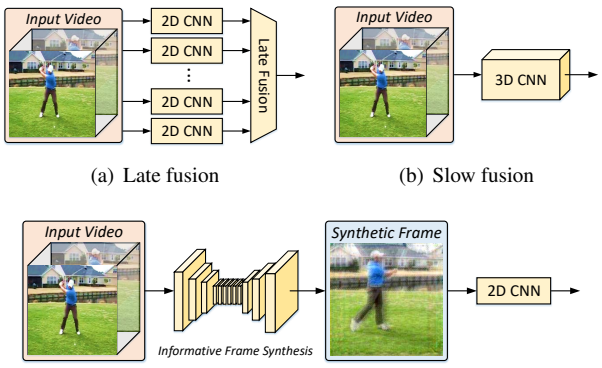
Zhaofan Qiu[†], Ting Yao[†], Yan Shu[§], Chong-Wah Ngo[‡], and Tao Mei[†]
[†] JD AI Research, Beijing, China
[§] University of Science and Techology of China, Hefei, China [‡] Singapore Management University, Singapore
{zhaofanqiu, tingyao.ustc, shuy.ustc}@gmail.com, cwngo@smu.edu.sg, tmei@jd.com

Abstract

Video is complex due to large variations in motion and rich content in fine-grained visual details. Abstracting useful information from such information-intensive media requires exhaustive computing resources. This paper studies a two-step alternative that first condenses the video sequence to an informative "frame" and then exploits off-the-shelf image recognition system on the synthetic frame. A valid question is how to define "useful information" and then distill it from a video sequence down to one synthetic frame. This paper presents a novel Informative Frame Synthesis (IFS) architecture that incorporates three objective tasks, i.e., appearance reconstruction, video categorization, motion estimation, and two regularizers, i.e., adversarial learning, color consistency. Each task equips the synthetic frame with one ability, while each regularizer enhances its visual quality. With these, by jointly learning the frame synthesis in an end-to-end manner, the generated frame is expected to encapsulate the required spatio-temporal information useful for video analysis. Extensive experiments are conducted on the large-scale Kinetics dataset. When comparing to baseline methods that map video sequence to a single image, IFS shows superior performance. More remarkably, IFS consistently demonstrates evident improvements on image-based 2D networks and clip-based 3D networks, and achieves comparable performance with the state-of-the-art methods with less computational cost.

1. Introduction

Recently, the development of Convolutional Neural Networks (CNN) convincingly demonstrates high capability of CNN in image-domain visual recognition. For instance, an ensemble of residual nets [9] achieves 3.5% top-5 error on the ImageNet test set, which is even lower than 5.1% of the reported human-level performance. Nevertheless, it is not trivial to apply a 2D CNN for video recognition. Since video is a temporal sequence with large variations and complexities, performing 2D CNN on individual frame cannot model temporal evolution across frames.



(a) Late fusion (b) Slow fusion

(c) Early fusion by frame synthesis

Figure 1. Modeling temporal evolution for video recognition by (a) aggregating 2D representations from sampled frames by **late fusion**, (b) **slow fusing** the input frames by 3D CNN and (c) condensing the input video to one frame as **early fusion**.

Extensive progresses have been made to model the temporal sequence for video recognition. These works can be grouped into three categories according to which stage temporal information is aggregated, as shown in Figure 1. The first one is **late fusion** [12, 20, 27, 38, 44, 45], which first extracts the image representation from 2D CNN for each frame and then aggregates the feature sequence for video recognition. Despite being straightforward by employing 2D CNN on video data, pixel-level temporal evolution over frames is overlooked. The second is **slow fusion** [2, 6, 10, 12, 21, 22, 23, 24, 31, 33], which feeds entire video clip into a network (e.g., 3D CNN) for spatio-temporal convolution. This type of networks builds temporal connections among pixels across space and time at the expense of computational cost. For example, the de facto ResNet-101 [9] requires 10G floating-number operations (FLOPs) for single crop on image data. When transferring this backbone to 3D CNN for 128-frame clip, the number of FLOPs is increased to 234G for SlowFast networks [6].

This paper addresses a new direction that **early fuses** the information from video sequence to one synthetic frame. Despite being a 2D image, the frame captures motion dy-

namics and visual details of a sequence. In this way, 2D CNN can be employed to learn both visual appearance as well as temporal evolution from just one frame. To this end, the crucial issue of early fusion becomes what information deserves to be preserved in a synthetic frame. Therefore, we propose a novel Informative Frame Synthesis (IFS) architecture guiding the generation of synthetic frames by multiple pre-defined tasks and regularizers. This architecture mainly consists of three components, i.e., a convolutional encoder-decoder network to transfer the input video sequence to a synthetic frame, three objective tasks to ensure that the frame is reconstructable, semantically and dynamically consistent with the original sequence, and two regularizers to preserve the fine-grained visual details. By jointly optimizing the transformation network in an end-to-end manner, the generated frame will attempt to encapsulate the required information of each task. In this way, the frame captures temporal evolution for video applications.

The main contribution of this work can be summarized as followings. First, IFS network is novelly proposed to learn the transformation from 3D video clips to 2D image frame. Second, different objectives and regularizers are proposed to encapsulate motion dynamics and visual details in a 2D frame. Extensive experiments are conducted on Kinetics dataset. Ablation studies investigate the impact of each task and regularizer towards frame synthesis. The results demonstrate that IFS with 2D CNN (e.g., ResNet-101) achieves comparable performance to more computationally expensive 3D CNN (e.g., I3D [2]). When the synthetic frames are stacked as a video summary for 3D CNN classification, higher performance is attained than most of the existing works with less computation.

2. Related Work

Video recognition has attracted intensive research interests in recent years due to its importance in different application areas, such as video surveillance, indexing, retrieval and robotics. We briefly group the methods for video recognition into two categories: hand-crafted feature-based and deep learning-based methods.

Early progresses are mostly based on the classifiers trained on **hand-crafted feature**, which usually starts by detecting spatio-temporal interest points and then describing them with local representations. Examples of hand-crafted feature include Space-Time Interest Points (STIP) [16], Histogram of Gradient and Histogram of Optical Flow [17], 3D HOG [14], SIFT-3D [26], Extended SURF [40], and improved dense trajectory [34, 35]. These hand-crafted descriptors are not particularly optimized for the video recognition task and may lack discriminative capacity.

The most recent approaches for video recognition are to devise **deep architectures** for end-to-end representation learning. Karparthy *et al.* stack CNN-based frame-level representations in a fixed size of windows and then leverage spatio-temporal convolutions for video categorization [12]. Benefiting from the usage of optical flow, in [27], the famous two-stream architecture is devised by applying two 2D CNN architectures separately on visual frames and stacked optical flows. Following the solution of two-stream networks, various schemes have been developed including convolutional fusion [7], key-volume mining [49], temporal segment networks [38] and temporal linear encoding [3]. To overcome the limitation of performing 2D CNN on modeling long-term dependencies, LSTM-RNN is proposed by Ng *et* al. [45] to model long-range temporal dynamics in videos. The aforementioned approaches are limited by treating video as a sequence of frames and optical flow for video understanding. More concretely, pixel-level temporal evolution across consecutive frames are not explored. The problem is addressed by 3D CNN proposed by Ji *et al.* [10], which directly learns spatio-temporal representation from a short video clip. Later in [31], Tran *et al.* devise a widely adopted 3D CNN, namely C3D, for supervised learning of video representation over 16-frame video clips using large-scale video datasets. Furthermore, performance of the 3D CNN is further boosted by inflated 2D kernels [2], decomposed 3D kernels [18, 21, 33], SlowFast networks [6, 41] and depth-wise 3D convolutions [4, 5, 32].

Our work also falls into the category of deep architecture learning, but in a direction that is seldom explored. The proposed network aims to condense video sequence into a synthetic frame summarizing spatio-temporal evolution. An early work DI [1] generates single "dynamic image" for each video by rank pooling technique to capture the temporal evolution. This pooling mechanism is improved in SVMP [36] by multiple instance learning context and decision boundaries in SVM. More recently, AWSD [30] distills the video sequence to single image by weighted pooling the surrounding pixels in a local window with adaptive duration. These three works transform the video clip to single image by manually designed formulation and are not learnable. The most closely related work is AVD [29], which generates single image from video sequence by 3D CNN with adversarial learning. Different from [29], our work focuses on the design of objective tasks and regularizers that can capture discriminative spatio-temporal characteristics in a single frame for recognition.

3. Informative Frame Synthesis (IFS)

IFS is essentially a generative model to synthesize a single frame that can infer visual and motion dynamics of a video clip. Figure 2 depicts the overview of IFS. We begin this section by presenting the problem formulation, followed by definition of tasks and regularizers. An end-to-end learning combining the loss functions is subsequently presented to train IFS. Finally, we explore different ways

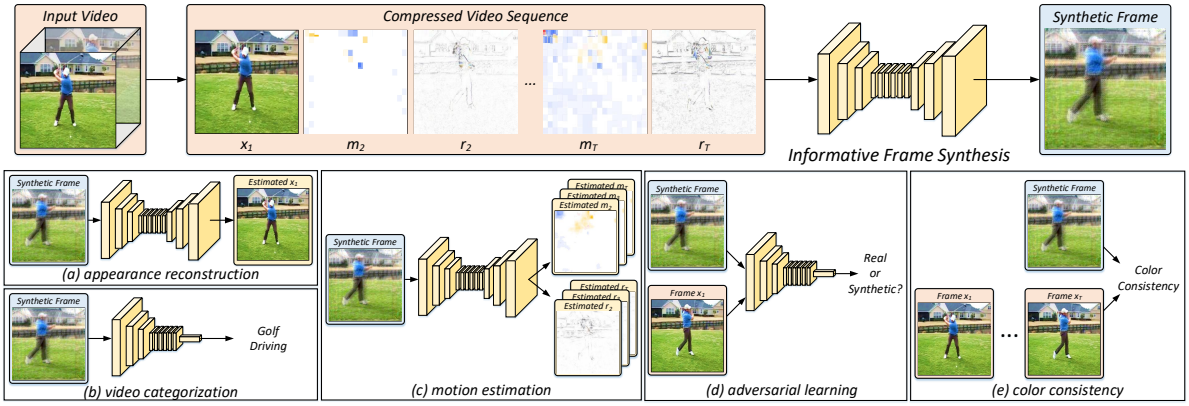


Figure 2. The three main components of Information Frame Synthesis (IFS): the convolution encoder-decoder network for transferring of video clip to a synthetic frame (upper); three objectives driving 3D to 2D transformation (lower left); two regularizers improving the visual quality of synthetic frame (lower right).

of utilizing IFS, including generating synthetic frames as a summary of long videos, for video recognition.

3.1. Problem Formulation

Denote a video clip with T frames as $\mathcal{X} = \{x_t | t = 1, ..., T\}$, where each frame is in resolution of $H \times W$ and has C color channels. Let IFS as a convolutional encoder-decoder network of function $\mathcal{F}$. Then, the problem of 2D frame synthesis is $\hat{x} = \mathcal{F}(\mathcal{X})$ where $\hat{x} \in \mathbb{R}^{C \times H \times W}$ is the 2D synthetic frame. As successive frames are expected to be visually similar, we follow the typical way of video compression by presenting a frame as its difference w.r.t to a key frame. Formally, take the first frame x_1 as the key frame, each subsequent frame x_t can be viewed as a distorted key frame of x_1 by motion vector m_t plus a residual r_t:

$$x_t = x_1 \circ m_t + r_t \ , \tag{1}$$

where $\circ$ denotes the pixel-level movement along the motion vector. In video compression, x_1 is named I-frame (intra-coded frame), and $\{m_t, r_t\}$ is the difference between the I-frame and a P-frame (predictive frame). Thus the video sequence $\mathcal{X}$ is reformulated as a compressed video sequence $\{x_1, m_2, r_2, ..., m_T, r_T\}$ as in [42, 47], which can be directly extracted from the videos encoded with standards such as MPEG-4, H.264, HEVC, etc.

3.2. Tasks for Synthetic Frame

A synthetic frame $\hat{x}$ is expected to capture the most essential information required for video recognition. This includes the abilities of (1) reconstructing the appearance of input clip $\mathcal{X}$ from $\hat{x}$, (2) predicting the semantic category of $\mathcal{X}$, if available during training time, based on $\hat{x}$, and (3) estimating the temporal dynamics over frames. To this end, we optimize the encoder-decoder network $\mathcal{F}$ by jointly learning the following three tasks.

Appearance Reconstruction. The synthetic frame $\hat{x}$ only contains $1/T$ digits compared with input clip. One straightforward way for video data distillation is to make the network $\mathcal{F}$ reversible. In other words, the input clip $\mathcal{X}$ can be recovered from the synthetic frame $\hat{x}$. In this task, we mainly focus on reconstructing the appearance of key frame. Considering the frame representation in compressed format as in Equ. (1), only the first frame (x_1) of $\mathcal{X}$ is retained while other frames contain only the motion coefficients and residuals to predict x_t. Therefore, we design another convolutional encoder-decoder network $\mathcal{F}_a^{-1}$ aiming at recovering the first frame x_1 from $\hat{x}$. The objective function is given by the mean squared error (MSE) between x_1 and the recovered frame as

$$\mathcal{L}_{app}(\mathcal{F}, \mathcal{F}_a^{-1}) = \left\| x_1 - \mathcal{F}_a^{-1}(\mathcal{F}(\mathcal{X})) \right\|_2^2 \ , \tag{2}$$

where $\|\cdot\|_2^2$ denotes the mean squared $L2$ norm across the entire frame. The reconstruction loss enforces $\hat{x}$ to recall the initial appearance of $\mathcal{X}$.

Video Categorization. Video label characterizes the spatio-temporal content of a video. Here, we employ a convolutional encoder network $\mathcal{C}$ trying to predict the video category from synthetic frame. Given the supervised pair of video clip and label $\{\mathcal{X}, y\}$, the cross entropy loss measures the deviation between the ground truth and the predicted label from network $\mathcal{C}$ as

$$\mathcal{L}_{cat}(\mathcal{F}, \mathcal{C}) = \text{CrossEntropy}\{y, \mathcal{C}(\mathcal{F}(\mathcal{X}))\} \ , \tag{3}$$

where $\mathcal{C}(\mathcal{F}(\mathcal{X}))$ is the probability of each category after being normalized to $(0, 1)$ by softmax operation. It is worth noticing that, among all the tasks and regularizers, video categorization is the only task that needs manual labels for supervised learning.

Motion Estimation. Optical flow is usually extracted to describe the displacement between two consecutive frames, representing a transformation function that warps one frame to another. Similar in spirit, we try to empower the synthetic frame with the capacity to transfer the first input frame x_1

to the t-th frame x_t. Benefited from the compressed representation in Equ. (1), this capacity is equivalent to reconstruct the motion vector m_t and the residual r_t. These maps capture only the changes of video based on the information in I-frame. We design an identical convolutional encoder-decoder network $\mathcal{F}_m^{-1}$ to predict all the motion vectors and residuals from the synthetic frame $\hat{x}$ in one feed-forward propagation. Then, the loss for motion estimation task is calculated by the averaged MSE between the estimated motion vector/residual and the real motion vector/residual as

$$\begin{aligned} &\{\hat{m}_2, \hat{r}_2, ..., \hat{m}_T, \hat{r}_T\} = \mathcal{F}_m^{-1}(\mathcal{F}(\mathcal{X})), \\ &\mathcal{L}_{mot}(\mathcal{F}, \mathcal{F}_m^{-1}) = \frac{1}{T-1}\sum_{t=2}^{T} \|m_t - \hat{m}_t\|_2^2 + \|r_t - \hat{r}_t\|_2^2 . \end{aligned} \tag{4}$$

Please note that there are $T-1$ motion vector maps and $T-1$ residual maps in total, altogether forming more digits than the synthetic frame. Such information "bottleneck" architecture will attempt to summarize the temporal dynamic across the entire sequence in a synthetic frame $\hat{x}$ by reducing inter-frame redundancy.

3.3. Regularizers for Synthetic Frame

The aforementioned three tasks drive the learning of the synthetic frame to recapitulate the appearance, semantic and motion information of input clip. From a different perspective, two regularizers are designed to enhance the visual quality of the generated frame.

Adversarial Learning. The first regularizer is derived from adversarial learning to force the synthetic frame to be visually similar to a real frame. A convolutional encoder network $\mathcal{D}$ is exploited as a discriminator to differentiate between the real and synthetic frames, while the network $\mathcal{F}$ is trained to maximally fool the discriminator by generating high-quality synthetic frames. The design follows generative adversarial learning [8] that trains two models, i.e., a generative model and a discriminative model, by pitting them against each other. The adversarial principle provides guidance for the frame synthesis by making the texture, pattern and structure between the real and synthetic frames indistinguishable by the discriminator. Formally, given the generated synthetic frame $\mathcal{F}(\mathcal{X})$ and the I-frame x_1, we calculate the adversarial loss as

$$\begin{aligned} \mathcal{R}_{adv}(\mathcal{D}) &= \|\mathcal{D}(x_1)\|_2^2 + \|1 - \mathcal{D}(\mathcal{F}(\mathcal{X}))\|_2^2 , \\ \mathcal{R}_{adv}(\mathcal{F}) &= \|\mathcal{D}(\mathcal{F}(\mathcal{X}))\|_2^2 , \end{aligned} \tag{5}$$

where $\mathcal{D}(\cdot)$ denotes the score to measure the reality of a frame by discriminator network $\mathcal{D}$. The loss function used in Equ. (5) is the least-square GAN in [48], which performs more stably in joint training. Similar to the standard GANs, the training of adversarial learning in IFS is a minmax game between $\mathcal{F}$ and $\mathcal{D}$, expecting an good equilibrium that $\mathcal{F}$ can produce a "realistic" synthetic frame after convergence.

Color Consistency. The second regularizer considers the meaning of each channel in a synthetic frame. Take the input video with RGB color space as an example, the number of channels C is equal to 3 for the input and synthetic frame. However, the meaning of each channel and the correlation between channels in the synthetic frame are usually not constrained. Not surprisingly, without this constraint, we observe that the network $\mathcal{F}$ will easily converge to generate an unpredictable and stochastic color space. The hue information of video content will be lost in that color space. Therefore, we propose an efficient way to enhance the color information in the synthetic frame by minimizing the distance between the average RGB value in the input video and that in the synthetic frame. Specifically, the color consistency loss is defined as

$$\mathcal{R}_{color}(\mathcal{F}) = \frac{1}{T}\sum_{x_t \in \mathcal{X}} \|\text{Ave}(x_t) - \text{Ave}(\mathcal{F}(\mathcal{X}))\|_2^2 , \tag{6}$$

where $\text{Ave}(\cdot) \in \mathbb{R}^C$ denotes the mean value of each channel averaged across $H \times W$ positions inside the frame.

3.4. Optimization

The overall training objective function of IFS integrates the losses from three tasks and two regularizers. The synthetic frame generation network $\mathcal{F}$ is updated as

$$\mathcal{L} = \mathcal{L}_{app} + \mathcal{L}_{cat} + \mathcal{L}_{mot} + \mathcal{R}_{adv} + \mathcal{R}_{color} , \tag{7}$$

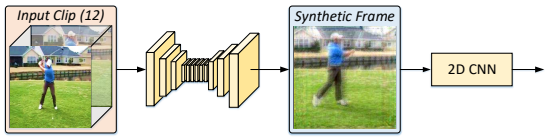
where the five losses are accumulated equally without weighting. Simultaneously, $\mathcal{F}_a^{-1}$, $\mathcal{C}$ and $\mathcal{F}_m^{-1}$ for the three tasks and D for the adversarial regularizer are jointly optimized with $\mathcal{F}$ for their respective objectives.

3.5. Video Recognition with Synthetic Frame

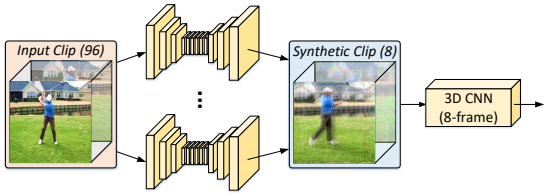
We develop several video recognition frameworks by employing 2D CNN and 3D CNN respectively on the synthetic frame. Figure 3 illustrates the two video recognition frameworks based on two typical networks.

(i) *Synthetic frame + 2D CNN:* The first one is simply by building a 2D CNN for the classification of each synthetic frame. Take $T = 12$ (i.e., one I-frame plus 11 P-frames) as an example, the 2D CNN on synthetic frame plays a similar role as a 12-frame 3D CNN. We employ ResNet-101 [9] pre-trained on ImageNet dataset [25] as the 2D classifier. We refer the 2D CNN as the default video classifier in our experiments unless otherwise stated.

(ii) *Synthetic clip + 3D CNN:* IFS can summarize a lengthy video into a short clip for video classification. Taking a 96-frame video as an example, IFS generates 8 consecutive synthetic frames, where each frame summarizes a 12-frame clip with a non-overlapping sliding window. In this case, we extend the 2D ResNet-101 to a 8-frame 3D CNN by the



(a) Synthetic frame + 2D CNN



(b) Synthetic clip + 3D CNN

Figure 3. Two examples of video recognition framework by integrating frame synthesis as early fusion.

strategy in [6, 21, 33] that inserts one $3 \times 1 \times 1$ temporal convolution after each spatial convolution. Similar as the SlowFast networks [6], we do not perform temporal downsampling in our 3D CNN, which means that the temporal dimension of each feature map is fixed as 8.

As shown in Figure 3, by early fusion of frames, IFS either condenses a video clip into one frame for 2D CNN classification or summarizes a clip as a synopsis of 8 frames for 3D CNN classification. The reduction is as much as 12 times of the original video length, which greatly decreases the computational cost.

IFS can also be degenerated to focus on capturing motion dynamics only by excluding the task of appearance construction. We name this variant as **IFS-mot**, where the produced synthetic frame only summarizes the information not included in x_1. Note that, with regularizers, the synthetic frame will still be visually sensible, or otherwise the 2D CNN pre-trained on ImageNet cannot be utilized for recognition. The performance of video recognition framework with different network architectures, or with different IFS variants, will be evaluated in the experimental section.

4. Experiments

4.1. Datasets

The experiments are conducted on Kinetics-400 [2], UCF101 [28] and HMDB51 [15] datasets. IFS is optimized on the training set of Kinetics-400 and then is applied on all the dataset. **Kinetics-400** is one of the large-scale action recognition benchmarks. It consists of around 300K videos from 400 action categories. The 300K videos are divided into 240K, 20K, 40K for training, validation and test set, respectively. Each video in this dataset is 10-second short clip cropped from the raw YouTube video. Note that the labels for test set are not publicly available and the performances on Kinetics-400 dataset are all reported on the validation set. After optimization, the effectiveness of synthetic frame is validated on Kinetics-400, UCF101 and HMDB51 in the context of video recognition. **UCF101** and **HMDB51** are two of the most popular video action recognition benchmarks. UCF101 consists of 13,320 videos from 101 action categories, and HMDB51 consists of 6,849 videos from 51 action categories. Each split in UCF101 includes about 9.5K training and 3.7K test videos, while a HMDB51 split contains 3.5K training and 1.5K test videos. We report the average results over three splits on these two datasets.

Table 1. The detailed architectures of encoder and encoder-decoder networks in IFS.

Layer	Encoder Network	Encoder-Decoder Network	Output Size
conv1	4×4, s=2	7×7	*En* : 64×112^2 *En-De* : 64×224^2
conv2	4×4, s=2	4×4, s=2	*En* : 128×56^2 *En-De* : 128×112^2
conv3	4×4, s=2	4×4, s=2	*En* : 256×28^2 *En-De* : 256×56^2
conv4	4×4, s=2	$\begin{bmatrix} 3 \times 3 \\ 3 \times 3 \end{bmatrix}_{\text{res}} \times 9$	*En* : 512×14^2 *En-De* : 256×56^2
conv5	4×4, s=2	4×4, s=1/2 4×4, s=1/2	*En* : 512×7^2 *En-De* : 64×224^2

4.2. Network Architecture

The detailed structures of encoder-decoder networks ($\mathcal{F}$, $\mathcal{F}_a^{-1}$, $\mathcal{F}_m^{-1}$) and encoder networks ($\mathcal{C}$, $\mathcal{D}$) are given in Table 1. The encoder-decoder networks originate from the 9-block ResNet in [48], which shows promising results on image-to-image translation task. This network contains two down-scale convolutional layers, two up-scale deconvolutional layers and nine residual blocks. For the encoder network, we devise the N-Layer classifier in [48], which stacks five 4×4 convolutional layers. Therefore, the encoder network can produce a feature map with 7×7 resolution which can be utilized to categorize the synthetic frame ($\mathcal{C}$) or distinguish between the real frame and synthetic frame ($\mathcal{D}$).

4.3. Training and Inference Strategy

The proposed IFS is implemented on PyTorch framework with multiple GPUs in parallel. We use MPEG-4 encoded videos, which have on average 11 P-frames for every I-frame. For the **training of IFS** (Section 3.4), we set the clip size as $T \times 224 \times 224$, where $T = 12$. The clips are randomly cropped without overlapping and is resized with the short edge in $[256, 340]$. IFS synthesizes one frame for average 12-frame clip. The clips are randomly flipped along horizontal direction for data augmentation. The network parameters are optimized by Adam [13] method with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The learning rate is initially set to

Table 2. Top-1 classification accuracy on Kinetics-400 with the synthetic frames end-to-endly generated by IFS variants. The variants are trained with the task of appearance reconstruction plus different combinations of regularizers. The last column "jpeg" shows the performance when the synthetic frames are compressed as JPEG images.

Task	Regularizer	$\mathcal{L}_{app}$	Top-1	
			end-to-end	jpeg
app	–	0.002	71.7	59.2
	adv	0.014	71.2	70.3
	$adv + color$	0.014	**72.8**	**72.8**

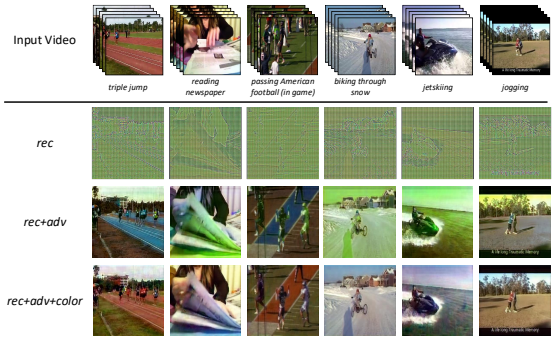


Figure 4. The examples of synthetic frames generated by IFS trained with the reconstruction task and different regularizers.

0.001, which is annealed down to zero following a cosine decay. The size of mini-batch is 128 and the optimization will be completed after 64 epoches. The optimized network $\mathcal{F}$ will be utilized for frame synthesis. For **video recognition** (Section 3.5), we utilize the same data pre-processing. The initial learning rate is 0.01 and the weights are optimized by standard stochastic gradient descent. Each mini-batch contains either 256 frames for 2D CNN or 256 clips for 3D CNN and the training completes after 256 epochs. In the inference stage, we employ the three-crop strategy in [6] and average the scores from 20/10 uniformly sampled synthetic frames/clips for 2D/3D CNN, respectively. All the variants of IFS follow exactly the same training and inference strategy as IFS.

4.4. Evaluations on the Regularizers

We first examine the impact of two regularizers, i.e., adversarial loss (adv) and color consistency ($color$) on IFS training. For simplicity, the IFS is only trained with the reconstruction task, i.e. Equ. (2). Table 2 summarizes the top-1 classification accuracies along with the loss when different regularizers are trained together. Figure 4 further contrasts the synthetic frames to show the impact of regularizers on visual effect. Without regularizer, the quality of synthetic frames is inexplicable despite the low value in reconstruction loss. The top-1 accuracy achieves 71.7% when a synthetic frame, represented as a matrix of floating point values, is input to 2D CNN. As expected, the accuracy decreases to 59.2% when these synthetic frames are compressed as JPEG images. By taking adversarial learning (adv) into account, the synthetic frames are more visually realistic. The accuracy does not drop dramatically when the frames are compressed as JPEG images. Nevertheless, this results in an increase of reconstruction loss and a drop of classification accuracy on the frame without compression. When color consistency is further considered, the accuracy reaches 72.8% for both the original and compressed synthetic frames. While a synthetic frame is not necessarily required to be visually realistic, a frame with decent visual quality can fully leverage the pre-trained network learnt from images (ImageNet) for video recognition. Making the synthetic frame visually similar to the real frame can ensure the effective transfer learning of the pre-trained network, especially when the input images are compressed to save storage space. In the rest of the paper, we compress the synthetic frames as JPEG images to reduce the demand for disk space. The time complexity of IFS and disk space consumption of storing JPEG images will be discussed in the supplementary material.

Table 3. The top-1 accuracy on Kinetics-400 of video classification with different combinations of tasks.

Task	$\mathcal{L}_{app}$	$\mathcal{L}_{cat}$	$\mathcal{L}_{mot}$	Top-1
app	0.014	–	–	72.8
cat	–	3.263		68.2
mot	–	–	0.007	71.6
$app + cat$	0.017	3.652	–	73.7
$app + mot$	0.015	–	0.010	74.5
$cat + mot$	–	3.340	0.008	72.8
$app + cat + mot$	0.015	3.701	0.011	**75.0**

4.5. Evaluations on the Tasks

Next, we study how video classification is affected by different tasks. Table 3 details the losses of the optimization on different tasks and the top-1 video classification accuracy. Figure 6 shows the examples of the input video sequences and their synthetic frames via different tasks. Note that both regularizers are trained together with the tasks. Among the three tasks, the reconstruction task exhibits the highest performance and the result is much better than the top-1 accuracy attained by the video categorization task. This somewhat reveals the weakness of frame distillation via categorization task, where the emphasis is on the image-level semantics rather than the pixel-level supervision as in the appearance reconstruction task and motion estimation task. Further improvement in classification is noted when combining multiple tasks for joint training. The highest result is attained when all the three tasks are involved in training. Figure 5 visualizes the frames synthesized by IFS for various videos, along with the reconstructed I-frames, esti-

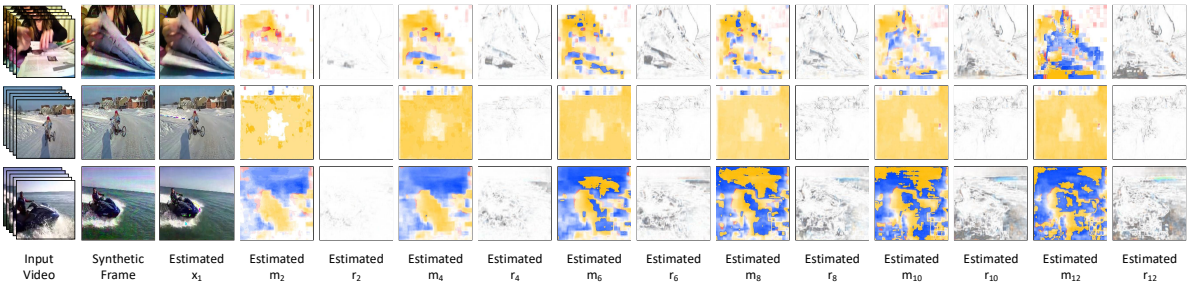


Figure 5. Examples of the input video, the synthetic frame and the estimated I-frame, motion vectors and residuals.

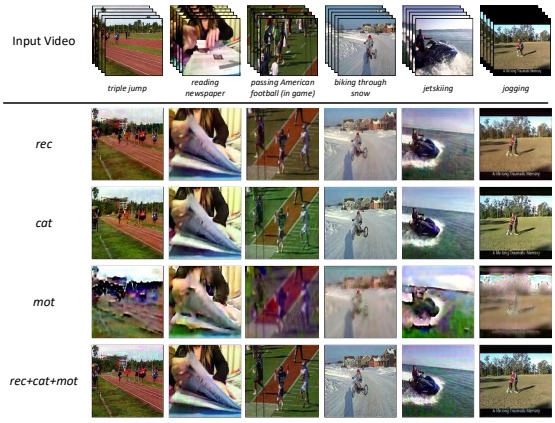


Figure 6. Examples of the synthetic frames generated by different tasks from the input sequences.

mated motion maps and residuals. The result verifies that the motion dynamics and visual details can be encapsulated simultaneously in a 2D frame by our IFS framework.

4.6. Evaluations on Informative Frame Synthesis

Next, we verify that the rich information (i.e., visual details and motion dynamics) captured in a 2D synthetic frame is helpful for video classification. We compare against the results when the raw data such as I-frames, motion vectors and residuals are directly extracted from MPEG videos for classification. Furthermore, other approaches such as optical flow, AVE, DI and SVMP are also compared as shown in Table 4. The **optical flow** modality firstly computes the value of optical flow between consecutive frames with [46], and then converts to a 2-channel image by scaling the value to [0, 255]. **AVE** is the average "frame" over all the frames. **DI** and **SVMP** applies approximate rank pooling [1] and SVM pooling [36], respectively, on the raw pixels of the frames, and takes decision boundary as the informative frame. AVE, DI and SVMP are all utilized to transform a 12-frame clip to a single frame. **IFS** and **IFS-mot** executes our framework with all three tasks and without appearance reconstruction task, respectively.

Table 4 lists the top-1 accuracy of video classification on Kinetics-400. Overall, the results across one-stream and two-stream evaluations consistently indicate that IFS leads to a performance boost against other baselines. An interesting observation is that DI and SVMP are superior to AVE, but still inferior to I-frame. We speculate that this is due to the lack of local structure and details of the output frame by AVE, DI and SVMP. In contrast, by encapsulating visual details and temporal evolution in 2D frame synthesis, IFS exhibits better performance. Further improvement can be attained if fusing the synthetic frame with either I-frame or optical flow. The result shows that the synthetic frame indeed complements well with other modalities. Note that when the appearance reconstruction task is not included in IFS (i.e., IFS-mot), the performance is also considerably better than the other motion-only modalities such as optical flow and motion vector.

Table 4. Performance comparisons on Kinetics-400 of video classification with the informative frame obtained via different ways.

Method	**One-stream**	**Two-stream**	
		+I-frame	**+flow**
I-frame	72.7	–	74.6
motion vector	34.0	73.4	65.2
residual	69.2	74.3	73.1
optical flow	63.2	74.6	–
AVE	70.2	73.7	73.8
DI [1]	71.6	74.9	72.9
SVMP [36]	71.0	74.4	72.3
IFS	**75.0**	**75.4**	**77.3**
IFS-mot	72.8	74.8	74.2

4.7. Comparison with the State-of-the-art

We compare with several state-of-the-art architectures in the context of video classification on Kinetics-400 dataset and Table 5 summarizes the performance comparison. When using 2D CNN for video classification, IFS and IFS+IFS-mot show better performance than I-frame and I-frame+flow, respectively, which directly exploit the intra-coded frame or optical flow in the sequence. Our IFS also outperforms AVD R101 [29] which synthesizes the frame by solely capitalizing on adversarial learning. Such result basically indicates the advantage of exploring multi-task learning plus two regularizers in IFS. Furthermore, IFS with less GFLOPs is even superior to several 3D CNN, e.g., I3D [2] and R(2+1)D [33], which spend about ten

Table 5. Performance comparison with the state-of-the-art methods on Kinetics-400, in terms of accuracy and computational complexity measured in GFLOPs × views. The number of views represents the number of clips sampled from the full video during inference, i.e., the number of temporal samples × the number of spatial crops. "N/A" indicates the numbers are not available for us.

Method	GFLOPs×views	top-1	top-5
STM R50 [11]	66×30	73.7	91.6
DFB R152 [19]	N/A×200	74.3	91.4
AVD R101 [29]	10×N/A	69.9	92.5
I3D [2]	108×N/A	72.1	90.3
R(2+1)D [33]	152×115	72.0	90.0
S3D-G [43]	143×N/A	77.2	93.0
Non-local R101 [39]	359×30	77.7	93.3
LGD-3D [23]	195×30	79.4	94.4
DFB R152-3D [19]	N/A×N/A	78.8	93.6
irCSN [32]	96.7×30	79.0	93.5
X3D-XL [5]	48.4×30	79.1	93.9
SlowFast 8×8 [6]	106×30	77.9	93.2
SlowFast 16×8 [6]	213×30	78.9	93.5
SlowFast 16×8+NL [6]	234×30	79.8	93.9
Two-stream AVD R101 [29]	21×N/A	75.1	93.4
Two-stream I3D [2]	216×N/A	75.7	92.0
Two-stream R(2+1)D [33]	304×115	73.9	90.9
Two-stream LGD-3D [23]	390×30	81.2	95.2
I-frame	10×60	72.7	89.5
I-frame+flow	21×60	74.6	91.3
IFS	10×60	75.0	91.5
IFS+**IFS**-mot	21×60	77.2	92.9
IFS-3D	98×30	79.0	94.0
IFS-3D+**IFS**-mot-3D	196×30	80.5	94.9

times GFLOPs. When taking 3D CNN as the architecture for video recognition, IFS-3D with less computation exhibits better performance than S3D-G [43] and Non-local [39]. Despite sharing similar computational load, IFS-3D outperforms SlowFast 8×8 [6]. The performance of IFS-3D is below LGD-3D [23] but the computation of IFS-3D is only half of LGD-3D. The two-stream integration of IFS-3D+IFS-mot-3D reaches the top-1 accuracy of 80.5%, which is also higher than that of AVD R101, I3D, and R(2+1)D in two-stream mode.

We finally evaluate the transferability of video representation learnt by IFS plus video classification networks on UCF101 and HMDB51 datasets. Specifically, we first pre-train both IFS network and 3D CNN of IFS-3D and IFS-mot-3D on Kinetics-400 dataset, and then fine-tune on UCF101 and HMDB51. Following [37], we freeze the parameters of all Batch Normalization layers except for the first one and add an extra dropout layer with 0.9 dropout rate to reduce the effect of over-fitting. Table 6 shows the performance comparisons. With only RGB input, IFS-3D pre-trained on Kinetics leads to better performance than I3D and STM [11]. When fusing two IFS variants, IFS-3D+IFS-mot-3D, without optical flow extraction, performs better than the two-stream R(2+1)D, AVD and I3D.

Table 6. Comparison with state-of-the-art on UCF101&HMDB51.

Method	+Flow	+Kinetics	U101	H51
IDT [35]			86.4	61.7
Two-stream [27]	✓		88.0	59.4
TSN [37]	✓		94.2	69.4
I3D [2]		✓	95.4	74.5
S3D [43]		✓	96.8	75.9
LGD-3D [23]		✓	97.0	75.7
STM [11]		✓	96.2	72.2
AVD [29]	✓	✓	97.3	77.1
I3D [2]	✓	✓	97.9	80.2
R(2+1)D [33]	✓	✓	97.3	75.9
LGD-3D [23]	✓	✓	98.2	80.5
IFS-3D		✓	97.4	76.2
IFS-3D+**IFS**-mot-3D		✓	98.2	80.3

4.8. Run Time and Disk Space

We analysis the run time of IFS and disk space consumption for storing JPEG images on Kinetics-400 dataset. The experiments are conducted on a regular server (Intel Xeon 2.40GHz CPU and 256 GB RAM) with four NVidia V100 GPUs. The run time of IFS to generate the synthetic frames of the entire Kinetics-400 dataset is around 5 hours, which is very efficient. Moreover, the disk space consumption for storing JPEG images is reduced from 2TB to 332GB via early fusing the frames by IFS.

5. Conclusions

This paper explores knowledge distillation from video sequence to frame for activity recognition. Particularly, we study the problem from a novel viewpoint of early fusing a 3D video clip to a 2D informative frame. To materialize our idea, we have devised Informative Frame Synthesis (IFS) architecture which integrates three objective tasks and two regularizers. Each task and regularizer empowers a synthetic frame to capture specific information ranging from visual to motion for video classification. Extensive experiments conducted on Kinetics dataset validate each design in IFS. The results of video classification by IFS with 2D CNN or 3D CNN on three video datasets demonstrate a good compromise between classification and computation cost. Furthermore, the ability in abstracting the knowledge from video as just-one-frame is potentially a new paradigm of video processing. IFS has demonstrated the feasibility of using such synthesized frames for video understanding.

Acknowledgments. This work was supported by the National Key R&D Program of China under Grant No. 2020AAA0108600.

References

[1] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In *CVPR*, 2016. 2, 7

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1, 2, 5, 7, 8

[3] Ali Diba, Vivek Sharma, and Luc Van Gool. Deep temporal linear encoding networks. In *CVPR*, 2017. 2

[4] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *ECCV*, 2020. 2

[5] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 2, 8

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 1, 2, 5, 6, 8

[7] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 2

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 4

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 4

[10] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. on PAMI*, 35(1):221–231, 2013. 1, 2

[11] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV*, 2019. 8

[12] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1, 2

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[14] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 2

[15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 5

[16] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005. 2

[17] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[18] Dong Li, Zhaofan Qiu, Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. Representing videos as discriminative sub-graphs for action recognition. In *CVPR*, 2021. 2

[19] Brais Martinez, Davide Modolo, Yuanjun Xiong, and Joseph Tighe. Action recognition with spatial-temporal discriminative filter banks. In *ICCV*, 2019. 8

[20] Zhaofan Qiu, Ting Yao, and Tao Mei. Deep quantization: Encoding convolutional activations with deep generative model. In *CVPR*, 2017. 1

[21] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 1, 2, 5

[22] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, and Tao Mei. Optimization planning for 3d convnets. In *ICML*, 2021. 1

[23] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. In *CVPR*, 2019. 1, 8

[24] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xiao-Ping Zhang, Dong Wu, and Tao Mei. Boosting video representation learning with multi-faceted integration. In *CVPR*, 2021. 1

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 4

[26] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM MM*, 2007. 2

[27] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2, 8

[28] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human action classes from videos in the wild. *CRCV-TR-12-01*, 2012. 5

[29] Mohammad Tavakolian, Mohammad Sabokrou, and Abdenour Hadid. Avd: Adversarial video distillation. *arXiv preprint arXiv:1907.05640*, 2019. 2, 7, 8

[30] Mohammad Tavakolian, Hamed R Tavakoli, and Abdenour Hadid. Awsd: Adaptive weighted spatiotemporal distillation for video representation. In *ICCV*, 2019. 2

[31] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1, 2

[32] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, 2019. 2, 8

[33] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 1, 2, 5, 7, 8

[34] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 2

[35] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 2, 8

[36] Jue Wang, Anoop Cherian, Fatih Porikli, and Stephen Gould. Video representation learning using discriminative pooling. In *CVPR*, 2018. 2, 7

[37] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 8

[38] 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 Trans. on PAMI*, 2018. 1, 2

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 8

[40] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008. 2

[41] Chao-Yuan Wu, Ross Girshick, Kaiming He, Christoph Feichtenhofer, and Philipp Krahenbuhl. A multigrid method for efficiently training video models. In *CVPR*, 2020. 2

[42] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *CVPR*, 2018. 3

[43] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 8

[44] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. In *AAAI*, 2021. 1

[45] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 1, 2

[46] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, 2007. 7

[47] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *CVPR*, 2016. 3

[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 4, 5

[49] Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao. A key volume mining deep framework for action recognition. In *CVPR*, 2016. 2