AR-Net: Adaptive Frame Resolution for Efficient Action Recognition

Yue Meng[1], Chung-Ching Lin[1], Rameswar Panda[1], Prasanna Sattigeri[1], Leonid Karlinsky[1], Aude Oliva[3], Kate Saenko[1,2], and Rogerio Feris[1]

[1] IBM Research AI, MIT-IBM Watson AI Lab
[2] Boston University
[3] Massachusetts Institute of Technology

Abstract. Action recognition is an open and challenging problem in computer vision. While current state-of-the-art models offer excellent recognition results, their computational expense limits their impact for many real-world applications. In this paper, we propose a novel approach, called AR-Net (Adaptive Resolution Network), that selects on-the-fly the optimal resolution for each frame conditioned on the input for efficient action recognition in long untrimmed videos. Specifically, given a video frame, a policy network is used to decide what input resolution should be used for processing by the action recognition model, with the goal of improving both accuracy and efficiency. We efficiently train the policy network jointly with the recognition model using standard back-propagation. Extensive experiments on several challenging action recognition benchmark datasets well demonstrate the efficacy of our proposed approach over state-of-the-art methods.

Keywords: Efficient Action Recognition, Multi-Resolution Processing, Adaptive Learning

1 Introduction

Action recognition has attracted intense attention in recent years. Much progress has been made in developing a variety of ways to recognize complex actions, by either applying 2D-CNNs with additional temporal modeling [31,54,15] or 3D-CNNs that model the space and time dimensions jointly [49,7,25]. Despite impressive results on commonly used benchmark datasets, the accuracy obtained by most of these models usually grows proportionally with their complexity and computational cost. This poses an issue for deploying these models in many resource-limited applications such as autonomous vehicles and mobile platforms.

Motivated by these applications, extensive studies have been recently conducted for designing compact architectures [44,27,28,64,2] or compressing models [13,57,9,35]. However, most of the existing methods process all the frames in a given video at the same resolution. In particular, orthogonal to the design of compact models, the computational cost of a CNN model also has much to do with the input frame size. To illustrate this, let us consider the video in Figure 1,

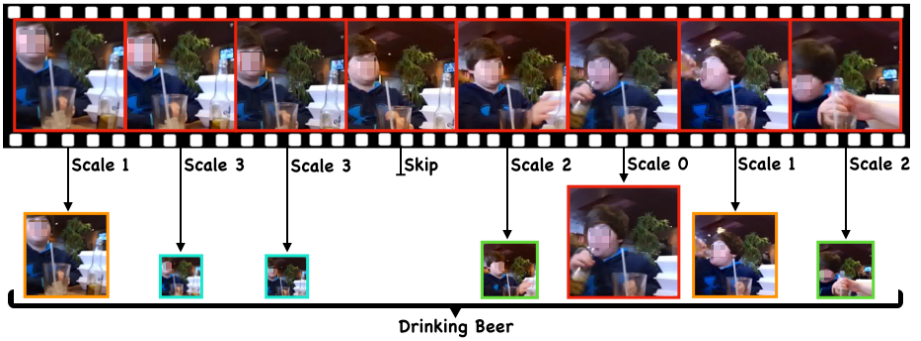

Figure 1. A conceptual overview of our approach. Rather than processing all the frames at the same resolution, our approach learns a policy to select the optimal resolution (or skip) per frame, that is needed to correctly recognize an action in a given video. As can be seen from the figure, the sixth frame is the most useful frame for recognition, therefore could be processed only with the highest resolution, while the rest of the frames could be processed at lower resolutions or even skipped without losing any accuracy. Best viewed in color.

represented by eight uniformly sampled frames. We ask, *Do all the frames need to be processed at the highest resolution (e.g., 224×224) to recognize the action as "Drinking Beer" in this video?* The answer is clear: No, the sixth frame is the most useful frame for recognition, therefore we could process only this frame at the highest resolution, while the rest of the frames could be processed at lower resolutions or even skipped (i.e., resolution set to zero) without losing any accuracy, resulting in large computational savings compared to processing all the frames with the same 224×224 resolution. Thus, in contrast to the commonly used one-size-fits-all scheme, we would like these decisions to be made individually per input frame, leading to different amounts of computation for different videos. Based on this intuition, we present a new perspective for efficient action recognition by adaptively selecting input resolutions, on a per frame basis, for recognizing complex actions.

In this paper, we propose AR-Net, a novel and differentiable approach to learn a decision policy that selects optimal frame resolution conditioned on inputs for efficient action recognition. The policy is sampled from a discrete distribution parameterized by the output of a lightweight neural network (referred to as the policy network), which decides on-the-fly what input resolution should be used on a per frame basis. As these decision functions are discrete and non-differentiable, we rely on a recent Gumbel Softmax sampling approach [29] to learn the policy jointly with the network parameters through standard back-propagation, without resorting to complex reinforcement learning as in [62,14,63]. We design the loss to achieve both competitive performance and resource efficiency required for action recognition. We demonstrate that adaptively selecting the frame resolution by a lightweight policy network yields not only significant savings in FLOPS (e.g., about 45% less computation over a state-of-the-art method [61] on

ActivityNet-v1.3 dataset [5]), but also consistent improvement in action recognition accuracy.

The main contributions of our work are as follows:

- We propose a novel approach that automatically determines what resolutions to use per target instance for efficient action recognition.
- We train the policy network jointly with the recognition models using backpropagation through Gumbel Softmax sampling, making it highly efficient.
- We conduct extensive experiments on three benchmark datasets (ActivityNet-V1.3 [5], FCVID [30] and Mini-Kinetics [7]) to demonstrate the superiority of our proposed approach over state-of-the-art methods.

2 Related Works

Efficient Action Recognition. Action recognition has made rapid progress with the introduction of a number of large-scale datasets such as Kinetics [7] and Moments-In-Time [39,40]. Early methods have studied action recognition using shallow classification models such as SVM on top of local visual features extracted from a video [34,53]. In the context of deep neural networks, it is typically performed by either 2D-CNNs [31,46,10,17,21] or 3D-CNNs [49,7,25]. A straightforward but popular approach is the use of 2D-CNNs to extract frame-level features and then model the temporal causality across frames using different aggregation modules such as temporal averaging in TSN [54], a bag of features scheme in TRN [65], channel shifting in TSM [36], depthwise convolutions in TAM [15], non-local neural networks [55], and LSTMs [12]. Many variants of 3D-CNNs such as C3D [49], I3D [7] and ResNet3D [25], that use 3D convolutions to model space and time jointly, have also been introduced for action recognition.

While extensive studies have been conducted in the last few years, limited efforts have been made towards *efficient* action recognition [62,61,20]. Specifically, methods for efficient recognition focus on either designing new lightweight architectures (e.g., R(2+1)D [51], Tiny Video Networks [44], channel-separated CNNs [50]) or selecting salient frames/clips conditioned on the input [63,62,33,20]. Our approach is most related to the latter which focuses on adaptive data sampling and is agnostic to the network architecture used for recognizing actions. Representative methods typically use Reinforcement Learning (RL) where an agent [62,14,63] or multiple agents [59] are trained with policy gradient methods to select relevant video frames, without deciding frame resolution as in our approach. More recently, audio has also been used as an efficient way to select salient frames for action recognition [33,20]. Unlike existing works, our framework requires neither complex RL policy gradients nor additional modalities such as audio. LiteEval [61] proposes a coarse-to-fine framework for resource efficient action recognition that uses a binary gate for selecting either coarse or fine features. In contrast, we address both the selection of optimal frame resolutions and skipping in an unified framework and jointly learn the selection and recognition mechanisms in a fully differentiable manner. Moreover, unlike binary sequential decision being made at every step in LiteEval, our proposed approach has the

flexibility in deciding multiple actions in a single step and also the scalability towards long untrimmed videos via multi-step skipping operations. We include a comprehensive comparison to LiteEval in our experiments.

Adaptive Computation. Many adaptive computation methods have been recently proposed with the goal of improving computational efficiency [3,4,52,56,23]. Several works have been proposed that add decision branches to different layers of CNNs to learn whether to exit the network for faster inference [18,38]. BlockDrop [60] effectively reduces the inference time by learning to dynamically select which layers to execute per sample during inference. Adaptive computation time for recurrent neural networks is also presented in [23]. SpotTune [24] learns to adaptively route information through finetuned or pre-trained layers. Reinforcement learning has been used to adaptively select different regions for fast object detection in large images [41,19]. While our approach is inspired by these methods, in this paper, we focus on adaptive computation in videos, where our goal is to adaptively select optimal frame resolutions for efficient action recognition.

Multi-Resolution Processing. Multi-resolution feature representations have a long history in computer vision. Traditional methods include image pyramids [1], scale-space representations [43], and coarse-to-fine approaches [42]. More recently, in the context of deep learning, multi-scale feature representations have been used for detection and recognition of objects at multiple scales [6,37], as well as to speed up deep neural networks [37,8]. Very few approaches have explored multi-scale recognition for efficient video understanding. A two-branch network that fuses the information of high-resolution and low-resolution video frames is proposed in [31]. bLVNet-TAM [15] also uses a two-branch multi-resolution architecture based on the Big-Little Net model [8], while learning long-term temporal dependencies across frames. SlowFast Networks [16] rely on a similar two-branch model, but each branch encodes different frame rates (i.e., different temporal resolutions), as opposed to frames with different spatial resolutions. Unlike these methods, rather than processing video frames at multiple resolutions with specialized network branches, our approach determines optimal resolution for each frame, with the goal of improving accuracy and efficiency.

3 Proposed Method

Given a video dataset $\mathcal{D} = \{(V_i, y_i)\}_{i=1}^{N}$, where each video V_i contains frames with spatial resolution $3 \times H_0 \times W_0$ and is labelled from the predefined classes: $y_i \in \mathbb{Y} = \{0, 1, ..., C-1\}$, our goal is to create an adaptive selection strategy that decides, per input frame, which resolution to use for processing by the classifier $\mathcal{F} : \mathbb{V} \rightarrow \mathbb{Y}$ with the goal of improving accuracy and efficiency. To this end, we first present an overview of our approach in Section 3.1. Then, we show how we learn the decision policy using Gumbel Softmax sampling in Section 3.2. Finally, we discuss the loss functions used for learning the decision policy in Section 3.3.

3.1 Approach Overview

Figure 2 illustrates an overview of our approach, which consists of a policy network and backbone networks for classifying actions. The policy network contains

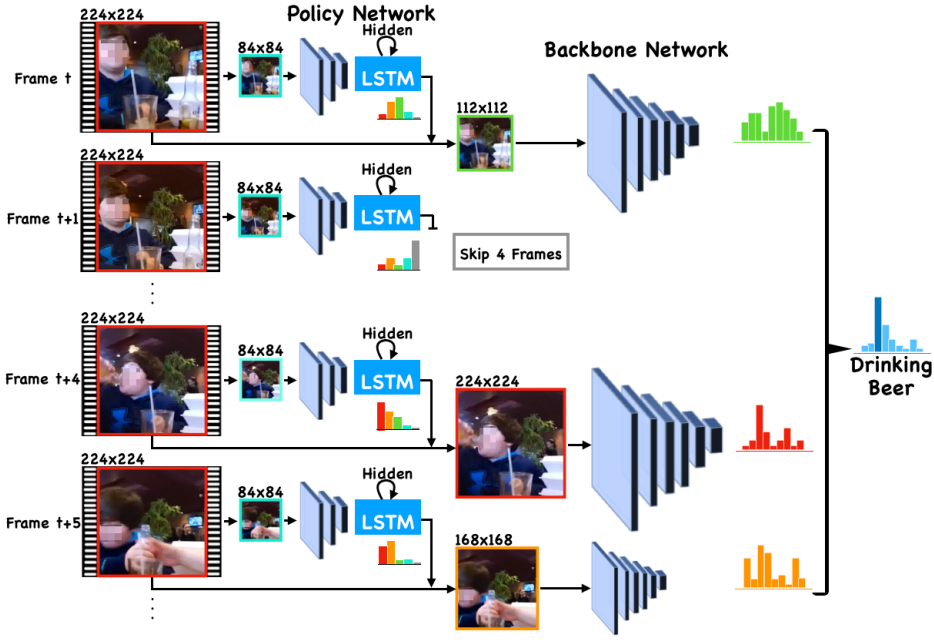

Figure 2. Illustration of our approach. AR-Net consists of a policy network and different backbone networks corresponding to different resolutions. The policy network decides what resolution (or skip) to use on a per frame basis to achieve accuracy and efficiency. In training, policies are sampled from a Gumbel Softmax distribution, which allows to optimize the policy network via backpropagation. During inference, input frames are first fed into the policy network to decide the proper resolutions, then the rescaled frames are routed to corresponding backbones to generate predictions. Finally the network averages all the predictions for action classification. Best viewed in color.

a lightweight feature extractor and an LSTM module that decides what resolutions (or skipping) to use per input frame, for efficient action recognition. Inspired by the compound scaling method [48], we adopt different network sizes to handle different resolutions, as a frame with a higher resolution should be processed by a heavier network because of its capability to handle the detailed visual information and vice versa.Furthermore, it is often unnecessary and inefficient to process every frame in a video due to large redundancy coming from static scenes or the frame quality being very low (blur, low-light condition, etc). Thus, we design a skipping mechanism in addition to the adaptive selection of frame resolutions in an unified framework to skip frames (i.e., resolution set to zero) whenever necessary to further improve the efficiency in action recognition.

During training, the policy network is jointly trained with the recognition models using Gumbel Softmax sampling, as we will describe next. At test time, an input frame is first fed into a policy network, whose output decides the proper resolutions, and then the resized frames are routed to the corresponding models to generate the predictions. Finally, the network averages all the predictions as

the action classification result. Note that the additional computational cost is incurred by resizing operations and the policy network, which are negligible in comparison to the original recognition models (the policy network is designed to be very lightweight, e.g., MobileNetv2 in our case).

3.2 Learning the Adaptive Resolution Policy

Adaptive Resolution. AR-Net adaptively chooses different frame scales to achieve efficiency. Denote a sequence of resolutions in descending order as $\{s_i\}_{i=0}^{L-1}$, where $s_0 = (H_0, W_0)$ stands for the original (also the highest) frame resolution, and $s_{L-1} = (H_{L-1}, W_{L-1})$ is the lowest resolution. The frame at time t in the l^{th} scale (resolution $s_l = (H_l, W_l)$) is denoted as I_t^l. We consider skipping frames as a special case "choosing resolutions s_∞". We define the skippings sequence (ascending order) as $\{F_i\}_{i=0}^{M-1}$, where the i^{th} operation means to skip the current frame and the following $(F_i - 1)$ frames from predictions. The choices for resolutions and skippings formulate our action space Ω.

Policy Network. The policy network contains a lightweight feature extractor $\Phi(\cdot; \theta_\Phi)$ and an LSTM module. At time step $t < T$ we resize the frame I_t to the lowest resolution I_t^{L-1} (for efficiency) and send it to the feature extractor,

$$f_t = \Phi(I_t^{L-1}; \theta_\Phi) \tag{1}$$

where f_t is a feature vector and θ_Φ denotes learnable parameters (we use θ_{name} for the learnable parameters in the rest of this section). The LSTM updates hidden state h_t and outputs o_t using the extracted feature and previous states,

$$h_t, o_t = \text{LSTM}(f_t, h_{t-1}, o_{t-1}; \theta_{LSTM}) \tag{2}$$

Given the hidden state, the policy network estimates the policy distribution and samples the action $a_t \in \Omega = \{0, 1, ...L + M - 1\}$ via the Gumbel Softmax operation (will be discussed in the next section),

$$a_t \sim \text{GUMBEL}(h_t, \theta_G) \tag{3}$$

If $a_t < L$, we resize the frame to spatial resolution $3 \times H_{a_t} \times W_{a_t}$ and forward it to the corresponding backbone network $\Psi_{a_t}(\cdot; \theta_{\Psi_{a_t}})$ to get a frame-level prediction,

$$y_t^{a_t} = \Psi_{a_t}(I_t^{a_t}; \theta_{\Psi_{a_t}}) \tag{4}$$

where $I_t^{a_t} \in \mathbb{R}^{3 \times H_{a_t} \times W_{a_t}}$ is the resized frame and $y_t^{a_t} \in \mathbb{R}^C$ is the prediction. Finally, all the frame-level predictions are averaged to generate the video-level prediction y for the given video V.

When the action $a_t >= L$, the backbone networks will skip the current frame for prediction, and the following $(F_{a_t - L} - 1)$ frames will be skipped by the policy network. Moreover, to save the computation, we share the policy network for generating both policy and predictions for the lowest resolution, i.e., $\Psi_{L-1} = \Phi$[4].

[4] The notation here is for brevity. Actually, the output for Φ is a feature vector, whereas the output for Ψ_{L-1} is a prediction. In implementation, we use a fully connected layer after the feature vector to get the prediction

Training using Gumbel Softmax Sampling. AR-Net makes decisions about which resolutions (or skipping) to use per training example. However, the fact that the decision policy is discrete makes the network non-differentiable and therefore difficult to optimize via backpropagation. One common practice is to use a score function estimator (e.g., REINFORCE [58,22]) to avoid back-propagating through the discrete samples. However, due to the undesirable fact that the variance of the score function estimator scales linearly with the discrete variable dimension (even when a variance reduction method is adopted), it is slow to converge in many applications [61,29]. As an alternative, in this paper, we adopt Gumbel-Softmax Sampling [29] to resolve this non-differentiability and enable direct optimization of the discrete policy in an efficient way.

The Gumbel Softmax trick [29] is a simple and effective way to substitute the original non-differentiable sample from a discrete distribution with a differentiable sample from a corresponding Gumbel-Softmax distribution. Specifically, at each time step t, we first generate the logits $z \in \mathbb{R}^{L+M-1}$ from hidden states h_t by a fully-connected layer $z = \mathrm{FC}(h_t, \theta_{FC})$. Then we use Softmax to generate a categorical distribution $\boldsymbol{\pi}_t$,

$$\boldsymbol{\pi}_t = \left\{ p_i \,\middle|\, p_i = \frac{\exp(z_i)}{\sum_{j=0}^{L+M-1} \exp(z_j)} \right\} \tag{5}$$

With the Gumbel-Max trick [29], the discrete samples from a categorical distribution are drawn as follows:

$$\hat{p} = \arg\max_i (\log p_i + G_i), \tag{6}$$

where $G_i = -\log(-\log U_i)$ is a standard Gumbel distribution with U_i sampled from a uniform i.i.d distribution $Unif(0, 1)$. Due to the non-differentiable property of arg max operation in Equation 6, the Gumbel Softmax distribution [29] is thus used as a continuous relaxation to arg max. Accordingly, sampling from a Gumbel Softmax distribution allows us to backpropagate from the discrete samples to the policy network. Let $\hat{P}$ be a one hot vector $[\hat{P}_0, ..., \hat{P}_{L+M-1}]$:

$$\hat{P}_i = \begin{cases} 1, & \text{if } i = \hat{p} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The one-hot coding of vector $\hat{P}$ is relaxed to a real-valued vector P using softmax:

$$P_i = \frac{\exp((\log p_i + G_i)/\tau)}{\sum_{j=0}^{L+M-1} \exp((\log p_j + G_j)/\tau)}, \qquad i \in [0, ..., L+M-1] \tag{8}$$

where τ is a temperature parameter, which controls the 'smoothness' of the distribution P, as $\lim_{\tau \to +\infty} P$ converges to a uniform distribution and $\lim_{\tau \to 0} P$ becomes a one-hot vector. We set $\tau = 5$ as the initial value and gradually anneal it down to 0 during the training, as in [24].

To summarize, during the forward pass, we sample the decision policy using Equation 6 (this is equivalent to the process mentioned in Equation 3 and $\theta_{FC} = \theta_G$) and during the backward pass, we approximate the gradient of the discrete samples by computing the gradient of the continuous softmax relaxation in Equation 8.

3.3 Loss Functions

During training, we use the standard cross-entropy loss to measure the classification quality as:

$$\mathcal{L}_{acc} = \mathbb{E}_{(V,y)\sim\mathcal{D}_{train}} \left[-y \log(\mathcal{F}(V;\Theta))\right] \tag{9}$$

where $\Theta = \{\theta_\Phi, \theta_{LSTM}, \theta_G, \theta_{\Psi_0}, ..., \theta_{\Psi_{L-2}}\}$ and (V, y) is the training video sample with associated one-hot encoded label vector. The above loss only optimizes for accuracy without taking efficiency into account. To address computational efficiency, we compute the GFLOPS for each individual module (and specific resolution of frames) offline and formulate a lookup table. We estimate the overall runtime GFLOPS for our network based on the offline lookup table $\text{GFLOPS}_\mathcal{F} : \Omega \to \mathbb{R}^+$and online policy $a_{V,t}$ for each training video $(V, y) \sim \mathcal{D}_{train}$. We use the GFLOPS per frame as a loss term to punish for high-computation operations,

$$\mathcal{L}_{flops} = \mathbb{E}_{(V,y)\sim\mathcal{D}_{train}} \left[\frac{1}{T}\sum_{t=0}^{T-1} \text{FLOPS}_\mathcal{F}(a_{V,t})\right] \tag{10}$$

Furthermore, to encourage the policy learning to choose more frames for skipping, we add an additional regularization term to enforce a balanced policy usage,

$$\mathcal{L}_{uni} = \sum_{i=0}^{L+M-1} \left(\mathbb{E}_{(V,y)\sim\mathcal{D}_{train}} \left[\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{1}(a_{V,t} = i)\right] - \frac{1}{L+M}\right)^2 \tag{11}$$

where $\mathbb{1}(\cdot)$ is the indicator function. Here $\mathbb{E}_{(V,y)\sim\mathcal{D}_{train}} \left[\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{1}(a_{V,t} = i)\right]$ represents the frequency of action i being made through the dataset. Intuitively, this loss function term drives the network to balance the policy usage in order to obtain a high entropy for the action distribution. To sum up, our final loss function for the training becomes:

$$\mathcal{L} = (1-\alpha)\cdot\mathcal{L}_{acc} + \alpha\cdot\mathcal{L}_{flops} + \beta\cdot\mathcal{L}_{uni} \tag{12}$$

where α denotes respective loss weight for the computing efficiency, and β controls the weight for the regularization term.

4 Experiments

In this section, we conduct extensive experiments to show that our model outperforms many strong baselines while significantly reducing the computation budget. We first show that our model-agnostic AR-Net boosts the performance of existing 2D CNN architectures (ResNet [26], EfficientNet [48]) and then show our method outperforms the State-of-the-art approaches for efficient video understanding. Finally, we conduct comprehensive experiments on ablation studies and qualitative analysis to verify the effectiveness of our policy learning.

4.1 Experimental Setup

Datasets. We evaluate our approach on three large-scale action recognition datasets: ActivityNet-v1.3 [5], FCVID(Fudan-Columbia Video Dataset) [30] and Mini-Kinetics [32]. ActivityNet [5] is labelled with 200 action categories and contains 10,024 videos for training and 4,926 videos for validation with an average duration of 117 seconds. FCVID [30] has 91,223 videos (45,611 videos for training and 45,612 videos for testing) with 239 label classes and the average length is 167 seconds. Mini-Kinetics dataset contains randomly selected 200 classes and 131,082 videos from Kinetics dataset [32]. We use 121,215 videos for training and 9,867 videos for testing. The average duration is 10 seconds.

Implementation Details. We uniformly sample $T = 16$ frames from each video. During training, images are randomly cropped to 224×224 patches with augmentation. At the inference stage, the images are rescaled to 256×256 and center-cropped to 224×224. We use four different frame resolutions ($L = 4$) and three skipping strategies ($M = 3$) as the action space. Our backbones network consists of ResNet-50 [26], ResNet-34 [26], ResNet-18 [26], and MobileNetv2 [45], corresponding to the input resolutions 224×224, 168×168, 112×112, and 84×84 respectively. The MobileNetv2 [45] is re-used and combined with a single-layer LSTM (with 512 hidden units) to serve as the policy network. The policy network can choose to skip 1, 2 or 4 frames.

Policy learning in the first stage is extremely sensitive to initialization of the policy. We observe that optimizing for both accuracy and efficiency is not effective with a randomly initialized policy. Thus, we divide The training process into 3 stages: warm-up, joint-training and fine-tuning. For warm-up, we fix the policy network and only train the backbones network (pretrained from ImageNet [11]) for 10 epochs with learning rate 0.02. Then the whole pipeline is jointly trained for 50 epochs with learning rate 0.001. After that, we fix the policy network parameters and fine-tune the backbones networks for 50 epochs with a lower learning rate of 0.0005. We set the initial temperature τ to 5, and gradually anneal it with an exponential decay factor of -0.045 in every epoch [29]. We choose $\alpha = 0.1$ and $\beta = 0.3$ for the loss function and use SGD [47] with momentum 0.9 for optimization. We will make our source code and models publicly available.

Baselines. We compare with the following baselines and existing approaches:

- UNIFORM: averages the frame-level predictions at the highest resolution 224×224 from ResNet-50 as the video-level prediction.

- LSTM: updates ResNet-50 predictions at the highest resolution 224×224 by hidden states and averages all predictions as the video-level prediction.
- RANDOM: uses our backbone framework but randomly samples policy actions from uniform distribution (instead of using learned policy distribution).
- Multi-Scale: gathers the frame-level predictions by processing different resolutions through our backbone framework (instead of selecting an optimal resolution with one corresponding backbone at each time step). This serves as a very strong baseline for classification, at the cost of heavy computation.
- AdaFrame [62]: uses MobileNetV2/ResNet-101 as lightweight CNN/backbone.
- LiteEval [61]: uses MobileNetV2/ResNet-101 as Policy Network/backbone.
- ListenToLook(Image) [20]: we compared with a variant of their approach with only the visual modality (MobileNetv2|ResNet-101). We also report their other results obtained by using audio data as an extra modality in the Figure 3.
- SCSampler [33]: as official code is not available, we re-implemented the SCSampler using AC loss as mentioned in [33]. We choose MobileNetv2 as the sampler network and use ResNet-50 as the backbone. We select 10 frames out of 16 frames for prediction, as in [33].

Metrics. We compute the mAP (mean average precision) and estimate the GFLOPS(gigabyte floating point operations per second) to reflect the performance for efficient video understanding. Ideally, a good system should have a high mAP with only a small amount of GFLOPS used during the inference stage. Since different baseline methods use different number of frames for classification, we calculate both GFLOPS per frame (denoted as GFLOPS/f) and GFLOPS per video (denoted as GFLOPS/V) in the following experiments.

4.2 Main Results

Adaptive Resolution Policy improves 2D CNN. We first compare our AR-Net with several simple baselines on ActivityNet and FCVID datasets to show how much performance our adaptive approach can boost in 2D convolution networks. We verify our method on both ResNet [26] and EfficientNet [48] to show the improvement is not limited to model architectures. As shown in Table 1, comparing to traditional "Uniform" and "LSTM" methods, we save 50% of the computation while getting a better classification performance.

We further show that it is the adaptively choosing resolutions and skippings that helps the most for efficient video understanding tasks. Taking ResNet architecture as an example, "Random Policy" can only reach 65.0% mAP on ActivityNet and 75.3% on FCVID, whereas AR-Net using learned policy can reach 73.8% and 81.3% respectively. Specifically, "Multi-Scale" can be a very strong baseline because it gathers all the predictions from multi-scale inputs through multiple backbones. It is noticeable that AR-Net's classification performance is comparable to the "Multi-Scale" baseline, while using 70% less computation. One possible explanation is that there exist noisy and misleading frames in the

videos, and AR-Net learns to skip those frames and uses the rest of the frames for prediction. Similar conclusion can also be drawn from using EfficientNet architectures, which shows our approach is model-agnostic.

Table 1: Action recognition results (in mAP and GFLOPS) on ActivityNet-v1.3 and FCVID. Our method consistently outperforms all simple baselines

Approach	Arch	ActivityNet-v1.3			FCVID		
		mAP(%)	GFLOPS/f	GFLOPS/V	mAP(%)	GFLOPS/f	GFLOPS/V
Uniform	ResNet	72.5	4.11	65.76	81.0	4.11	65.76
LSTM		71.2	4.12	65.89	81.1	4.12	65.89
Random Policy		65.0	**1.04**	**16.57**	75.3	**1.03**	**16.49**
Multi-Scale		73.5	6.90	110.43	**81.3**	6.90	110.43
AR-Net		**73.8**	2.09	33.47	81.3	2.19	35.12
Uniform	Efficient Net	78.8	1.80	28.80	83.5	1.80	28.80
LSTM		78.0	1.81	28.88	83.7	1.81	28.88
Random Policy		72.5	**0.38**	**6.11**	79.7	**0.38**	**6.11**
Multi-Scale		79.5	2.35	37.56	84.2	2.35	37.56
AR-Net		**79.7**	0.96	15.29	**84.4**	0.88	14.06

Table 2: Results for video classification on ActivityNet-v1.3 and FCVID datasets

Approach	ActivityNet-v1.3			FCVID		
	mAP(%)	GFLOPS/f	GFLOPS/V	mAP(%)	GFLOPS/f	GFLOPS/V
AdaFrame [62]	71.5	3.16	78.97	80.2	3.01	75.13
LiteEval [61]	72.7	3.80	95.10	80.0	3.77	94.30
ListenToLook(Image) [20]	72.3	5.09	81.36	-	-	-
SCSampler [33]	72.9	2.62	41.95	81.0	2.62	41.95
AR-Net(ResNet)	**73.8**	**2.09**	**33.47**	**81.3**	**2.19**	**35.12**
AR-Net(EfficientNet)	**79.7**	**0.96**	**15.29**	**84.4**	**0.88**	**14.06**

Adaptive Resolution Policy outperforms state-of-the-art methods. We compare the performance of AR-Net with several state-of-the-art methods on ActivityNet and FCVID in Table 2. The result section of the table is divided into two parts. The upper part contains all the methods using Residual Network architecture, whereas the lower part shows the best result we have achieved by using the latest EfficientNet [48] architecture. Usually it is hard to improve the classification accuracy while maintaining a low computation cost, but our "AR-Net(ResNet)" outperforms all the state-of-the-art methods in terms of mAP scores, frame-level GFLOPS and video-level GFLOPS. Our method achieves 73.8% mAP on ActivityNet and 81.3% mAP on FCVID while using 17% $\sim$ 64% less computation budgets compared with other approaches. This shows the power of our adaptive resolution learning approach in efficient video understanding tasks. When integrated with EfficientNet [48], our "AR-Net(EfficientNet)" further gains 5.9% in mAP on ActivityNet and 3.1% on FCVID, with 54%$\sim$60% less computation compared to "AR-Net(ResNet)". Since there is no published result using EfficientNet for efficient video understanding, these results can serve as the new baselines for future research.

Figure 3 illustrates the GFLOPS-mAP curve on ActivityNet dataset, where our AR-Net obtains significant computational efficiency and action recognition accuracy with much fewer GFLOPS than other baseline methods. We quote

the reported results on MultiAgent [59], AdaFrame[62] and ListenToLook[20] (here "(IA|R)" and "(MN|R)" are short for "(Image-Audio|ResNet-101)" and "(MobileNetV2|ResNet-101)" mentioned in [20]). The results of LiteEval [61] are generated through the codes shared by the authors, and the results of SCSampler [33] are obtained by our re-implementation following their reported details. ListenToLook (IA|R) denotes models using both visual and audio data as inputs. Given the same ResNet architectural family, our approach achieves substantial improvement compared to the best competitors, demonstrating the superiority of our method. Additionally, our best performing model, which employs EfficientNet [48] architecture, yields more than 5% improvement in mAP at the same computation budgets. It shows that our approach of adaptively selecting proper resolutions on a per frame basis is able to yield significant savings in computation budget and to improve recognition precision.

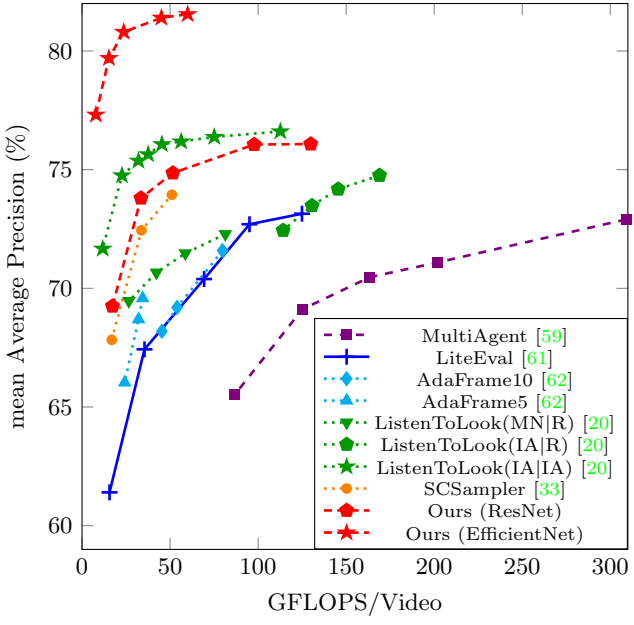

Figure 3. Comparisons with state-of-the-art alternatives on ActivityNet dataset. Our proposed AR-Net obtains the best recognition accuracy with much fewer GFLOPS than the compared methods. We directly quote the numbers reported in published papers when possible and compare the mAP against the average GFLOPs per test video. See text for more details.

Table 3: Results for video classification on Mini-Kinetics dataset

Approach	Mini-Kinetics		
	Top1(%)	GFLOPS/f	GFLOPS/V
LiteEval [61]	61.0	3.96	99.00
SCSampler [33]	70.8	2.62	41.95
AR-Net(ResNet)	**71.7**	**2.00**	**32.00**
AR-Net(EfficientNet)	**74.8**	**1.02**	**16.32**

Further Experiment on Mini-Kinetics. To further test the capability of our method, we conduct experiments on Mini-Kinetics dataset. Compared with the recent methods LiteEval [61] and SCSampler [33], our method achieves better Top-1 accuracy and the computation cost is reduced with noticeable margin. In brief, our method consistently outperform the existing methods in terms of

accuracy and speed on different datasets, which implies our AR-Net provides an effective framework for various action recognition applications.

4.3 Ablation Studies

Effectiveness of choosing resolution and skipping. Here we inspect how each type of operation enhances the efficient video understanding. We define three different action spaces: "Resolution Only" (the policy network can only choose different resolutions), "Skipping Only"(the policy network can only decide how many frames to skip) and "Resolution+Skipping". We follow the same training procedures as illustrated in Section 4.1 and evaluate each approach on ActivityNet dataset. We adjust the training loss to keep their GFLOPS at the same level and we only compare the differences in classification performances. As shown in Table 4, comparing with baseline methods ("Uniform" and "LSTM"), they all improve the performance, and the best strategy is to combine skippings and choosing resolutions. Intuitively, skipping frames can be seen as "choosing zero resolution" for the current frame, hence gives the flexibility for the network in making decisions.

Table 4: Results of different policy settings on ActivityNet-v1.3

Policy Settings	mAP(%)	GFLOPS/f	GFLOPS/V
Uniform	72.5	4.11	65.76
LSTM	71.2	4.12	65.89
Resolution Only	73.4	2.13	34.08
Skipping Only	72.7	2.21	34.90
Resolution+Skipping	**73.8**	**2.09**	**33.47**

Table 5: Results of different losses on ActivityNet-v1.3

Losses	α	β	mAP(%)	GFLOPS/f	GFLOPS/V
Acc	0.0	0.0	**74.5**	3.75	60.06
Acc+Eff	0.1	0.0	73.8	2.28	36.48
Acc+Eff+Uni	0.1	0.3	73.8	**2.09**	**33.47**

Trade-off between accuracy and efficiency. As discussed in Section 3.3, hyper-parameters α and β in Equation 12 affect the classification performance, efficiency and policy distribution. Here we train our model using 3 different weighted combinations: "Acc" (only using accuracy-related loss), "Acc+Eff"(using accuracy and efficiency losses) and "Acc+Eff+Uni"(using all the losses). As shown in Table 5, training with "Acc" will achieve the highest mAP, but the computation cost will be similar to "Uniform" method (GFLOPS/V=65.76). Adding the efficiency loss term will decrease the computation cost drastically, whereas training with "Acc+Eff+Uni" will drop the GFLOPS even further. One reason is that the network tends to skip more frames in the inference stage. Finally, we use hyper-parameters $\alpha = 0.1$, $\beta = 0.3$ in our training.

Different training strategies. We explore several strategies for training the adaptive learning framework. As shown in Table 6, the best practice comes from "Warm-Up+Joint+Finetuning" so we adopt it in training our models.

Table 6: Results of different training strategies on ActivityNet-v1.3

Training Strategy			mAP(%)	GFLOPS/f	GFLOPS/V
Warm-Up	Joint	Finetuning			
✗	✓	✗	67.1	1.16	17.86
✓	✓	✗	73.3	2.03	32.40
✓	✓	✓	73.8	2.09	33.47

4.4 Qualitative Analysis

An intuitive view of how AR-Net achieves efficiency is shown in Figure 4. We conduct experiments on ActivityNet-v1.3 and FCVID testing sets. Videos are uniformly sampled in 8 frames. The upper row of each example shows original input frames, and the lower row shows the frames processed by our policy network for predictions. AR-Net keeps the most indicative frames (e.g. Futsal and Fencing) in original resolution and resizes or skips frames that are irrelevant or in low quality (blurriness). After being confident about the predictions, AR-Net will avoid to use original resolution even informative contents appear again (e.g. last several frames in "Pitching a tent"/"Windsurfing"). The last two examples show that our approach is able to capture both object-interaction (clipper-dog) and background changes.

Figure 4. Qualitative examples from ActivityNet and FCVID. We uniformly sample 8 frames per video and AR-Net chooses the proper resolutions or skipping. Relevant frames are kept in original resolution whereas non-informative frames are resized to lower resolution or skipped for computation efficiency.

5 Conclusion

In this paper, we have demonstrated the power of adaptive resolution learning on a per frame basis for efficient video action recognition. Comprehensive experiments show that our method can work in a full range of accuracy-speed operating points, from a version that is both faster and more accurate than comparable visual-only models to a new, state-of-the-art accuracy-throughput version based on the EfficientNet [48] architecture. The proposed learning framework is model-agnostic, which allows applications to various sophisticated backbone networks and the idea can be generally adopted to explore other complex video understanding tasks.

Acknowledgement

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00341.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

1. Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid methods in image processing. RCA engineer **29**(6), 33–41 (1984)
2. Araujo, A., Negrevergne, B., Chevaleyre, Y., Atif, J.: Training compact deep learning models for video classification using circulant matrices. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
3. Bengio, E., Bacon, P.L., Pineau, J., Precup, D.: Conditional computation in neural networks for faster models. arXiv preprint arXiv:1511.06297 (2015)
4. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
5. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the ieee conference on computer vision and pattern recognition. pp. 961–970 (2015)
6. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: European conference on computer vision. pp. 354–370. Springer (2016)
7. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
8. Chen, C.F., Fan, Q., Mallinar, N., Sercu, T., Feris, R.: Big-little net: An efficient multi-scale feature representation for visual and speech recognition. arXiv preprint arXiv:1807.03848 (2018)
9. Chen, W., Wilson, J., Tyree, S., Weinberger, K., Chen, Y.: Compressing neural networks with the hashing trick. In: International conference on machine learning. pp. 2285–2294 (2015)
10. Chéron, G., Laptev, I., Schmid, C.: P-cnn: Pose-based cnn features for action recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 3218–3226 (2015)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2625–2634 (2015)

13. Dong, X., Huang, J., Yang, Y., Yan, S.: More is less: A more complicated network with less inference complexity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5840–5848 (2017)
14. Fan, H., Xu, Z., Zhu, L., Yan, C., Ge, J., Yang, Y.: Watching a small portion could be as good as watching all: Towards efficient video classification. In: IJCAI International Joint Conference on Artificial Intelligence (2018)
15. Fan, Q., Chen, C.F.R., Kuehne, H., Pistoia, M., Cox, D.: More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. In: Advances in Neural Information Processing Systems. pp. 2261–2270 (2019)
16. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6202–6211 (2019)
17. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spatiotemporal multiplier networks for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4768–4777 (2017)
18. Figurnov, M., Collins, M.D., Zhu, Y., Zhang, L., Huang, J., Vetrov, D., Salakhutdinov, R.: Spatially adaptive computation time for residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1039–1048 (2017)
19. Gao, M., Yu, R., Li, A., Morariu, V.I., Davis, L.S.: Dynamic zoom-in network for fast object detection in large images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6926–6935 (2018)
20. Gao, R., Oh, T.H., Grauman, K., Torresani, L.: Listen to look: Action recognition by previewing audio. arXiv preprint arXiv:1912.04487 (2019)
21. Gkioxari, G., Malik, J.: Finding action tubes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 759–768 (2015)
22. Glynn, P.W.: Likelihood ratio gradient estimation for stochastic systems. Communications of the ACM **33**(10), 75–84 (1990)
23. Graves, A.: Adaptive computation time for recurrent neural networks. arXiv preprint arXiv:1603.08983 (2016)
24. Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., Feris, R.: Spottune: transfer learning through adaptive fine-tuning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4805–4814 (2019)
25. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6546–6555 (2018)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
27. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
28. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)
29. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
30. Jiang, Y.G., Wu, Z., Wang, J., Xue, X., Chang, S.F.: Exploiting feature and class relationships in video categorization with regularized deep neural networks. IEEE transactions on pattern analysis and machine intelligence **40**(2), 352–364 (2017)

31. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)
32. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
33. Korbar, B., Tran, D., Torresani, L.: Scsampler: Sampling salient clips from video for efficient action recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6232–6242 (2019)
34. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
35. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710 (2016)
36. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7083–7093 (2019)
37. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
38. McGill, M., Perona, P.: Deciding how to decide: Dynamic routing in artificial neural networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 2363–2372 (2017)
39. Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S.A., Yan, T., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., et al.: Moments in time dataset: one million videos for event understanding. IEEE transactions on pattern analysis and machine intelligence **42**(2), 502–508 (2019)
40. Monfort, M., Ramakrishnan, K., Andonian, A., McNamara, B.A., Lascelles, A., Pan, B., Gutfreund, D., Feris, R., Oliva, A.: Multi-moments in time: Learning and interpreting models for multi-action video understanding. arXiv preprint arXiv:1911.00232 (2019)
41. Najibi, M., Singh, B., Davis, L.S.: Autofocus: Efficient multi-scale inference. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9745–9755 (2019)
42. Pedersoli, M., Vedaldi, A., Gonzalez, J., Roca, X.: A coarse-to-fine approach for fast deformable object detection. Pattern Recognition **48**(5), 1844–1853 (2015)
43. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on pattern analysis and machine intelligence **12**(7), 629–639 (1990)
44. Piergiovanni, A., Angelova, A., Ryoo, M.S.: Tiny video networks. arXiv preprint arXiv:1910.06961 (2019)
45. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
46. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
47. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: International conference on machine learning. pp. 1139–1147 (2013)

48. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)
49. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015)
50. Tran, D., Wang, H., Torresani, L., Feiszli, M.: Video classification with channel-separated convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5552–5561 (2019)
51. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6450–6459 (2018)
52. Veit, A., Belongie, S.: Convolutional networks with adaptive inference graphs. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–18 (2018)
53. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR 2011. pp. 3169–3176. IEEE (2011)
54. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016)
55. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
56. Wang, X., Yu, F., Dou, Z.Y., Darrell, T., Gonzalez, J.E.: Skipnet: Learning dynamic routing in convolutional networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 409–424 (2018)
57. Wen, W., Xu, C., Wu, C., Wang, Y., Chen, Y., Li, H.: Coordinating filters for faster deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 658–666 (2017)
58. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3-4), 229–256 (1992)
59. Wu, W., He, D., Tan, X., Chen, S., Wen, S.: Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6222–6231 (2019)
60. Wu, Z., Nagarajan, T., Kumar, A., Rennie, S., Davis, L.S., Grauman, K., Feris, R.: Blockdrop: Dynamic inference paths in residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8817–8826 (2018)
61. Wu, Z., Xiong, C., Jiang, Y.G., Davis, L.S.: Liteeval: A coarse-to-fine framework for resource efficient video recognition. In: Advances in Neural Information Processing Systems. pp. 7778–7787 (2019)
62. Wu, Z., Xiong, C., Ma, C.Y., Socher, R., Davis, L.S.: Adaframe: Adaptive frame selection for fast video recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1278–1287 (2019)
63. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2678–2687 (2016)
64. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018)

65. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 803–818 (2018)