

# Listen to Look: Action Recognition by Previewing Audio

Ruohan Gao<sup>1,2\*</sup>   Tae-Hyun Oh<sup>2†</sup>   Kristen Grauman<sup>1,2</sup>   Lorenzo Torresani<sup>2</sup>  
<sup>1</sup>The University of Texas at Austin   <sup>2</sup>Facebook AI Research  
 rhgao@cs.utexas.edu, {taehyun, grauman, torresani}@fb.com

## Abstract

*In the face of the video data deluge, today’s expensive clip-level classifiers are increasingly impractical. We propose a framework for efficient action recognition in untrimmed video that uses audio as a preview mechanism to eliminate both short-term and long-term visual redundancies. First, we devise an IMGAUD2VID framework that hallucinates clip-level features by distilling from lighter modalities—a single frame and its accompanying audio—reducing short-term temporal redundancy for efficient clip-level recognition. Second, building on IMGAUD2VID, we further propose IMGAUD-SKIMMING, an attention-based long short-term memory network that iteratively selects useful moments in untrimmed videos, reducing long-term temporal redundancy for efficient video-level recognition. Extensive experiments on four action recognition datasets demonstrate that our method achieves the state-of-the-art in terms of both recognition accuracy and speed.*

## 1. Introduction

With the growing popularity of portable image recording devices as well as online social platforms, internet users are generating and sharing an ever-increasing number of videos every day. According to a recent study, it would take a person over 5 million years to watch the amount of video that will be crossing global networks each month in 2021 [1]. Therefore, it is imperative to devise systems that can recognize actions and events in these videos both accurately and efficiently. Potential benefits extend to many video applications, including video recommendation, summarization, editing, and browsing.

Recent advances in action recognition have mostly focused on building powerful clip-level models operating on short time windows of a few seconds [55, 61, 17, 10, 71, 16]. To recognize the action in a test video, most methods densely apply the clip classifier and aggregate the prediction scores of all the clips across the video. Despite encouraging

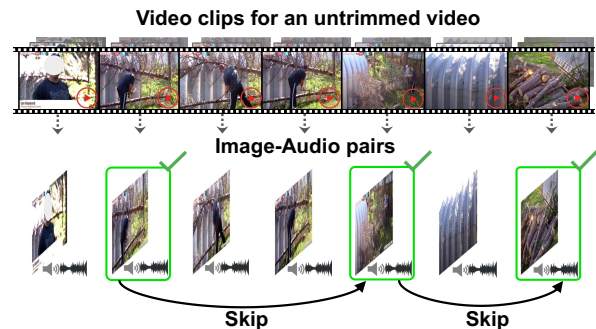


Figure 1: Our approach learns to use audio as an efficient preview of the accompanying visual content, at two levels. First we replace the costly analysis of video clips with a more efficient processing of image-audio pairs. A single image captures most of the appearance information within the clip, while the audio provides important dynamic information. Then our video skimming module selects the key moments (a subset of image-audio pairs) to perform efficient video-level action recognition.

progress, this approach becomes computationally impractical in real-world scenarios where the videos are untrimmed and span several minutes or even hours.

We contend that processing all frames or clips in a long untrimmed video may be unnecessary and even counter-productive. Our key insight is that there are two types of redundancy in video, manifested in both short-term clips as well as long-term periods. First, there is typically high temporal redundancy across the entire video (Fig. 1). Many clips capture the same event repetitively, suggesting it is unnecessary to process all the clips. Second, there is redundancy even within a clip: the visual composition within a short time span does not change abruptly; temporally adjacent frames are usually very similar, though there are temporal dynamics (motion) across frames. Therefore, it can be wasteful to process all clips and frames, especially when the video is very long. Moreover, for many activities, the actual actions taking place in the video can be very sparse. It is often a few important moments that are useful for recognition, while the rest actually distract the classifier. For example, in a typical video of surfing, a person might talk for a long

\*Work done during an internship at Facebook AI Research.

†T.-H. Oh is now with Dept. EE, POSTECH, Korea.

time and prepare the equipment before he/she begins to surf.

Our idea is to use *audio as an efficient video preview* to reduce both the clip-level and the video-level redundancy in long untrimmed videos. First, instead of processing a whole video clip, we propose an IMGAUD2VID teacher-student distillation framework to hallucinate a video descriptor (*e.g.*, an expensive 3D CNN feature vector) from a single video frame and its accompanying audio. Based on our lightweight image-audio network, we further propose a novel attention-based long short-term memory (LSTM) network, called IMGAUD-SKIMMING, which scans through the entire video and selects the key moments for the final video-level recognition. Both ideas leverage audio as a fast preview of the full video content. Our distilled image-audio model efficiently captures information over short extents, while the skimming module performs fast long-term modeling by skipping over irrelevant and/or uninformative segments across the entire video.

Audio has ideal properties to aid efficient recognition in long untrimmed videos: audio contains dynamics and rich contextual temporal information [23] and, most importantly, it is much more computationally efficient to process compared to video frames. For example, as shown in Fig. 1, within a short clip of the action chopping wood, a single frame includes most of the appearance information contained in the clip, *i.e.*, {person, axe, tree}, while the accompanying audio (the sound of the axe hitting the tree in this case) contains useful cues of temporal dynamics. Across the entire video, audio can also be beneficial to select the key moments that are useful for recognition. For example, the sound of the person talking initially can suggest that the actual action has not started, while the sound of the electric saw may indicate that the action is taking place. Our approach automatically learns such audio signals.

We experiment on four datasets (Kinetics-Sounds, MiniSports1M, ActivityNet, UCF-101) and demonstrate the advantages of our framework. Our main contributions are threefold. Firstly, we are the first to propose to replace the expensive extraction of clip descriptors with an efficient proxy distilled from audio. Secondly, we propose a novel video-skimming mechanism that leverages image-audio indexing features for efficient long-term modeling in untrimmed videos. Thirdly, our approach pushes the envelope of the trade-off between accuracy and speed favorably; we achieve state-of-the-art results on action recognition in untrimmed videos with few selected frames or clips.

## 2. Related Work

**Action Recognition.** Action recognition in video has been extensively studied in the past decades. Research has transitioned from initial methods using hand-crafted local spatiotemporal features [37, 72, 66] to mid-level descriptors [50, 30, 67], and more recently to deep video repre-

sentations learned end-to-end [55, 32, 17]. Various deep networks have been proposed to model spatiotemporal information in videos [61, 10, 49, 71, 16]. Recent work includes capturing long-term temporal structure via recurrent networks [81, 12] or ranking functions [18], pooling across space and/or time [69, 24], modeling hierarchical or spatiotemporal information in videos [48, 63], building long-term temporal relations [73, 85], or boosting accuracy by treating audio as another (late-fused) input modality [76, 41, 70, 34].

The above work focuses on building powerful models to improve recognition without taking the computation cost into account, whereas our work aims to perform efficient action recognition in long untrimmed videos. Some work balances the accuracy-efficiency trade-off by using compressed video representations [74, 54] or designing efficient network architectures [78, 89, 11, 62, 39]. In contrast, we propose to leverage audio to enable efficient clip-level and video-level action recognition in long untrimmed videos.

**Action Proposal and Localization.** The goal of action localization [31, 79, 53, 84] is to find the temporal start and end of each action within a given untrimmed video and simultaneously recognize the action class. Many approaches [8, 79, 40, 68] first use action proposals to identify candidate action segments. While reminiscent of our audio preview mechanism, the computational cost of most action proposal methods is several orders of magnitude larger. They generate a large number of clip proposals from the video, and then use flow [40] or deep features [8, 79]) for proposal selection. The selection stage is typically even more expensive than the final classification. Instead, our method addresses video classification, and high efficiency is a requirement in our design.

**Audio-Visual Analysis.** Recent work uses audio for an array of video understanding tasks outside of action recognition, including self-supervised representation learning [47, 5, 7, 45, 35, 59], audio-visual source separation [45, 2, 14, 20, 83, 22], localizing sounds in video frames [6, 52, 60], and generating sounds from video [46, 87, 21, 44, 86]. Different from all the work above, we focus on leveraging audio for efficient action recognition.

**Cross-modal Distillation.** Knowledge distillation [29] addresses the problem of training smaller models from larger ones. We propose to distill the knowledge from an expensive clip-based model to a lightweight image-audio based model. Other forms of cross-modal distillation consider transferring supervision from RGB to flow or depth [28] or from a visual network to an audio network, or vice versa [7, 47, 3, 19]. In the opposite direction of ours, DistInit [25] performs uni-modal distillation from a pre-trained image model to a video model for representation learning from unlabeled video. Instead, we perform multi-modal distillation from a video model to an image-audio model

for efficient clip-based action recognition.

### Selection of Frames or Clips for Action Recognition.

Our approach is most related to the limited prior work on selecting salient frames or clips for action recognition in untrimmed videos. Whereas we use only weakly labeled video to train, some methods assume strong human annotations, *i.e.*, ground truth temporal boundaries [80] or sequential annotation traces [4]. Several recent methods [57, 15, 77, 75] propose reinforcement learning (RL) approaches for video frame selection. Without using guidance from strong human supervision, they ease the learning process by restricting the agent to a rigid action space [15], guiding the selection process of the agent with a global memory module [77], or using multiple agents to collaboratively perform frame selection [75].

Unlike any of the above, we introduce a video skimming mechanism to select the key moments in videos aided by audio. We use audio as an efficient way to preview dynamic events for fast video-level recognition. Furthermore, our approach requires neither strong supervision nor complex RL policy gradients, which are often unwieldy to train. SCSampler [36] also leverages audio to accelerate action recognition in untrimmed videos. However, they only consider video-level redundancy by sampling acoustically or visually salient clips. In contrast, we address both clip-level and video-level redundancy, and we jointly learn the selection and recognition mechanisms. We include a comprehensive experimental comparison to methods in this genre.

**Video Summarization.** Video summarization work also aims to select keyframes or clips [38, 26, 42, 82], but with the purpose of conveying the gist of the video to a human viewer. Instead, our work aims to select features useful for activity recognition. Beyond the difference in goal, our iterative attention-based mechanism is entirely novel as a frame selection technique.

## 3. Approach

Our goal is to perform accurate and efficient action recognition in long untrimmed videos. We first formally define our problem (Sec. 3.1); then we introduce how we use audio as a clip-level preview to hallucinate video descriptors based on only a single static frame and its accompanying audio segment (Sec. 3.2); finally we present how we leverage image-audio indexing features to obtain a video-level preview, and learn to skip over irrelevant or uninformative segments in untrimmed videos (Sec. 3.3).

### 3.1. Problem Formulation

Given a long untrimmed video  $\mathcal{V}$ , the goal of video classification is to classify  $\mathcal{V}$  into a predefined set of  $C$  classes. Because  $\mathcal{V}$  can be very long, it is often intractable to process all the video frames together through a single deep network due to memory constraints. Most current

approaches [55, 32, 61, 10, 49, 62, 71, 16] first train a clip-classifier  $\Omega(\cdot)$  to operate on a short fixed-length video clip  $\mathbf{V} \in \mathbb{R}^{F \times 3 \times H \times W}$  of  $F$  frames with spatial resolution  $H \times W$ , typically spanning several seconds. Then, given a test video of arbitrary length, these methods densely apply the clip-classifier to  $N$  clips  $\{\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_N\}$  which are taken at a fixed hop size across the entire video. The final video-level prediction is obtained by aggregating the clip-level predictions of all  $N$  clips.

As discussed in Sec. 1, such paradigms for video recognition are highly inefficient at two levels: (1) *clip-level*—within each short clip  $\mathbf{V}$ , temporally close frames are visually similar, and (2) *video-level*—across all the clips in  $\mathcal{V}$ , often only a few clips contain the key moments for recognizing the action. Our approach addresses both levels of redundancy via novel uses of audio.

Each video clip  $\mathbf{V}$  is accompanied by an audio segment  $\mathbf{A}$ . The starting frame  $\mathbf{I}$  among the  $F$  frames within the short clip  $\mathbf{V}$  usually contains most of the appearance cues already, while the audio segment  $\mathbf{A}$  contains rich contextual temporal information (recall the wood cutting example in Fig. 1). Our idea is to replace the powerful but expensive clip-level classifier  $\Omega(\cdot)$  that takes  $F$  frames as input with an efficient image-audio classifier  $\Phi(\cdot)$  that only takes the starting frame  $\mathbf{I}$  and its accompanying audio segment  $\mathbf{A}$  as input, while preserving the clip-level information as much as possible. Namely, we seek to learn  $\Phi(\cdot)$  such that

$$\Omega(\mathbf{V}_j) \approx \Phi(\mathbf{I}_j, \mathbf{A}_j), \quad j \in \{1, 2, \dots, N\}, \quad (1)$$

for a given pre-trained clip-classifier  $\Omega(\cdot)$ . In Sec. 3.2, we design an IMGAUD2VID distillation framework to achieve this goal. Through this step, we replace the processing of high-dimensional video clips  $\{\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_N\}$  with a lightweight model analyzing compact image-audio pairs  $\{(\mathbf{I}_1, \mathbf{A}_1), (\mathbf{I}_2, \mathbf{A}_2), \dots, (\mathbf{I}_N, \mathbf{A}_N)\}$ .

Next, building on our efficient image-audio classifier  $\Phi(\cdot)$ , to address video-level redundancy we design an attention-based LSTM network called IMGAUD-SKIMMING. Instead of classifying every image-audio pair using  $\Phi(\cdot)$  and aggregating all their prediction results, our IMGAUD-SKIMMING framework iteratively selects the most useful image-audio pairs. Namely, our method efficiently selects a small subset of  $T$  image-audio pairs from the entire set of  $N$  pairs in the video (with  $T \ll N$ ) and only aggregates the predictions from these selected pairs. We present our video skimming mechanism in Sec. 3.3.

### 3.2. Clip-Level Preview

We present our approach to perform efficient clip-level recognition and our IMGAUD2VID distillation network architecture. As shown in Fig. 2, the clip-based model takes a video clip  $\mathbf{V}$  of  $F$  frames as input and based on that video volume generates a clip descriptor  $\mathbf{z}^{\mathbf{V}}$  of dimensionality  $D$ .

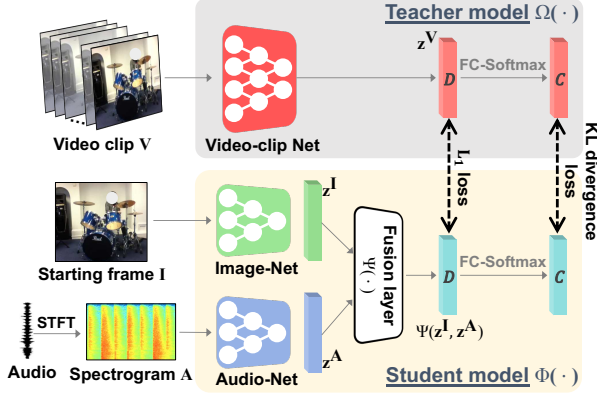


Figure 2: IMGAUD2VID distillation framework: The teacher model is a video-clip classifier, and the student model consists of a visual stream that takes the starting frame of the clip as input and an audio stream that takes the audio spectrogram as input. By processing only a single frame and the clip’s audio, we get an estimate of what the expensive video descriptor would be for the full clip.

A fully-connected layer is used to make predictions among the  $C$  classes in Kinetics. For the student model, we use a two-stream network: the image stream takes the first frame  $I$  of the clip as input and extracts an image descriptor  $z^I$ ; the audio stream takes the audio spectrogram  $A$  as input and extracts an audio feature vector  $z^A$ . We concatenate  $z^I$  and  $z^A$  to generate an image-audio feature vector of dimensionality  $D$  using a fusion network  $\Psi(\cdot)$  that consists of two fully-connected layers. A final fully-connected layer is used to produce a  $C$ -class prediction like the teacher model.

The teacher model  $\Omega(\cdot)$  returns a softmax distribution over  $C$  classification labels. These predictions are used as soft targets for training weights associated with the student network  $\Phi(\cdot)$  using the following objective:

$$\mathcal{L}_{KL} = - \sum_{\{(V, I, A)\}} \sum_c \Omega_c(V) \log \Phi_c(I, A), \quad (2)$$

where  $\Omega_c(V)$  and  $\Phi_c(I, A)$  are the softmax scores of class  $c$  for the teacher model and the student model, respectively. We further impose an  $\mathcal{L}_1$  loss on the clip descriptor  $z^V$  and the image-audio feature to regularize the learning process:

$$\mathcal{L}_1 = \sum_{\{(z^V, z^I, z^A)\}} \|z^V - \Psi(z^I, z^A)\|_1. \quad (3)$$

The final learning objective for IMGAUD2VID distillation is a combination of these two losses:

$$\mathcal{L}_{Dist.} = \mathcal{L}_1 + \lambda \mathcal{L}_{KL}, \quad (4)$$

where  $\lambda$  is the weight for the KL divergence loss. The training is done over the image and audio student networks (producing representations  $z^I$  and  $z^A$ , respectively) and the fusion model  $\Psi(\cdot)$  with respect to a fixed teacher video-clip model. The teacher model we use is a R(2+1)D-18 [62] video-clip classifier, which is pre-trained on Kinetics [33].

Critically, processing the audio for a clip is substantially faster than processing all its frames, making audio an efficient preview. See Sec. 4.1 for cost comparisons. After distillation, we fine-tune the student model on the target dataset to perform efficient clip-level action recognition.

### 3.3. Video-Level Preview

IMGAUD2VID distills knowledge from a powerful clip-based model to an efficient image-audio based model. Next, we introduce how we leverage the distilled image-audio network to perform efficient video-level recognition. Recall that for long untrimmed video, processing only a subset of clips is desirable both for speed and accuracy, *i.e.*, to ignore irrelevant content.

We design IMGAUD-SKIMMING, an attention-based LSTM network (Fig. 3), which interacts with the sequence of image-audio pairs  $\{(I_1, A_1), (I_2, A_2), \dots, (I_N, A_N)\}$ , whose features are denoted as  $\{z_1^I, z_2^I, \dots, z_N^I\}$  and  $\{z_1^A, z_2^A, \dots, z_N^A\}$ , respectively. At the  $t$ -th time step, the LSTM cell takes the *indexed* image feature  $\tilde{z}_t^I$  and the *indexed* audio feature  $\tilde{z}_t^A$ , as well as the previous hidden state  $h_{t-1}$  and the previous cell output  $c_{t-1}$  as input, and produces the current hidden state  $h_t$  and the cell output  $c_t$ :

$$h_t, c_t = \text{LSTM}(\Psi(\tilde{z}_t^I, \tilde{z}_t^A), h_{t-1}, c_{t-1}), \quad (5)$$

where  $\Psi(\cdot)$  is the same fusion network used in IMGAUD2VID with the same parameters. To fetch the indexed features  $\tilde{z}_t^I$  and  $\tilde{z}_t^A$  from the feature sequences, an indexing operation is required. This operation is typically non-differentiable. Instead of relying on approximating policy gradients as in prior work [15, 77, 75], we propose to deploy a differentiable soft indexing mechanism, detailed below.

We predict an image query vector  $q_t^I$  and an audio query vector  $q_t^A$  from the hidden state  $h_t$  at each time step through two prediction networks  $\text{Query}^I(\cdot)$  and  $\text{Query}^A(\cdot)$ . The query vectors,  $q_t^I$  and  $q_t^A$ , are used to query the respective sequences of image indexing features  $\{z_1^I, z_2^I, \dots, z_N^I\}$  and audio indexing features  $\{z_1^A, z_2^A, \dots, z_N^A\}$ . The querying operation is intended to predict which part of the untrimmed video is more useful for recognition of the action in place and decide where to “look at” and “listen to” next. It is motivated by attention mechanisms [27, 58, 65, 64], but we adapt this scheme to the problem of selecting useful moments for action recognition in untrimmed video.

Figure 4 illustrates our querying mechanism. First, we use one fully-connected layer  $\text{Key}(\cdot)$  to transform indexing features  $z$  to indexing keys  $k$ . Then, we get an attention score  $\frac{k^\top q}{\sqrt{d}}$  for each indexing key in the sequence, where  $d$  is the dimensionality of the key vector. A Softmax layer normalizes the attention scores and generates an attention weight vector  $w$  by:

$$w = \text{Softmax}\left(\frac{[k_1 k_2 \dots k_N]^\top \cdot q}{\sqrt{d}}\right), \quad (6)$$



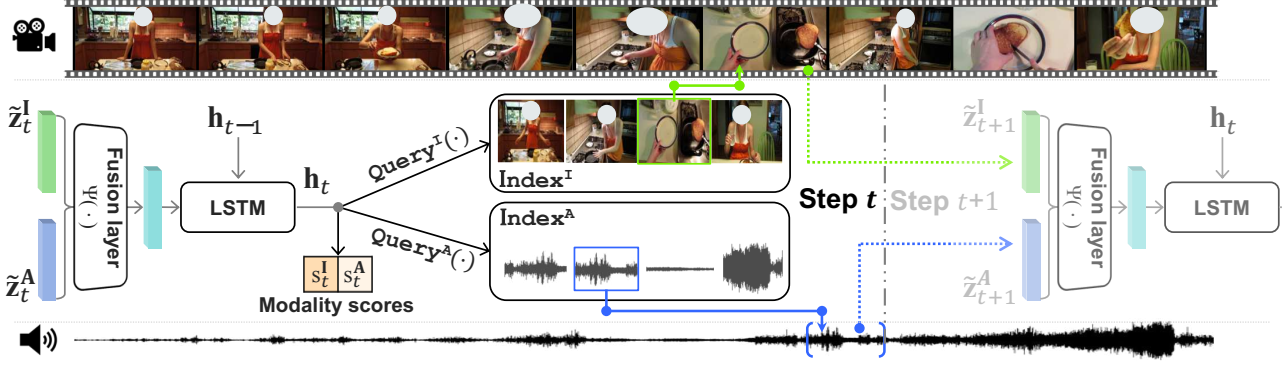


Figure 3: Our IMGAUD-SKIMMING network is an LSTM network that interacts with the sequences of image and audio indexing features to select where to “look at” and “listen to” next. At each time step, it takes the image feature and audio feature for the current time step as well as the previous hidden state and cell output as input, and produces the current hidden state and cell output. The hidden state for the current time step is used to make predictions about the next moment to focus on in the untrimmed video through the querying operation illustrated in Fig. 4. The average-pooled IMGAUD2VID features of all selected time steps is used to make the final prediction of the action in the video.

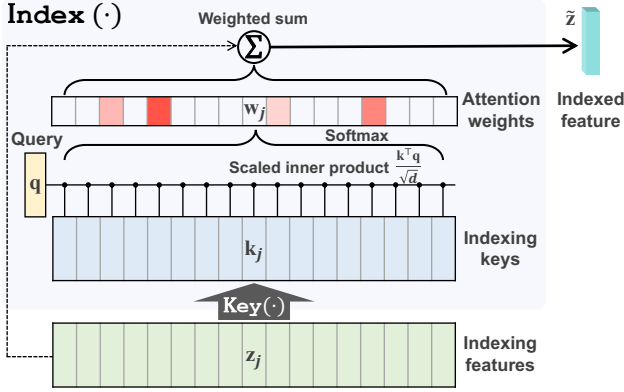


Figure 4: Attention-based frame selection mechanism.

where  $\mathbf{k}_j = \text{Key}(\mathbf{z}_j)$ ,  $j \in \{1, 2, \dots, N\}$ .

At each time step  $t$  (we omit  $t$  for simplicity if deducible), one could obtain the frame index for the next time step by  $\arg\max(\mathbf{w})$ . However, this operation is not differentiable. Instead of directly using the image and audio features of the selected frame index, we use the weighted average of the sequence of indexing features to generate an aggregated feature vector  $\tilde{\mathbf{z}}_{t+1}^I = \text{Index}^I(\mathbf{w}_t)$  and  $\tilde{\mathbf{z}}_{t+1}^A = \text{Index}^A(\mathbf{w}_t)$  as input to the fusion network  $\Psi(\cdot)$ , where

$$\begin{aligned} \text{Index}^I(\mathbf{w}) &:= \sum_{j=1}^N w_j \mathbf{z}_j^I, \\ \text{Index}^A(\mathbf{w}) &:= \sum_{j=1}^N w_j \mathbf{z}_j^A, \quad w_{j \in \{1, \dots, N\}} \in \mathbb{R}_+. \end{aligned} \quad (7)$$

The querying operations are performed independently on the visual and audio modalities, and produce distinct weight vectors  $\mathbf{w}_t^I$  and  $\mathbf{w}_t^A$  to find the visually-useful and acoustically-useful moments, respectively. These two weight vectors may give importance to different moments in the sequence. We fuse this information by dynamically adjusting how much to rely on each modality at each step.

To this end, we predict two modality scores  $s_t^I$  and  $s_t^A$ , from the hidden state  $\mathbf{h}_t$  through a two-way classification layer.  $s_t^I$  and  $s_t^A$  ( $s_t^I, s_t^A \in [0, 1]$ ,  $s_t^I + s_t^A = 1$ ) indicate how much the system decides to rely on the visual modality versus the audio modality, respectively, at time step  $t$ . Then, the image and audio feature vectors for the next time step are finally obtained by aggregating the feature vectors predicted both visually and acoustically, as follows:

$$\begin{aligned} \tilde{\mathbf{z}}_{t+1}^I &= s_t^I \cdot \text{Index}^I(\mathbf{w}_t^I) + s_t^A \cdot \text{Index}^I(\mathbf{w}_t^A), \\ \tilde{\mathbf{z}}_{t+1}^A &= s_t^I \cdot \text{Index}^A(\mathbf{w}_t^I) + s_t^A \cdot \text{Index}^A(\mathbf{w}_t^A). \end{aligned} \quad (8)$$

Motivated by iterative attention [43], we repeat the above procedure for  $T$  steps, and average the image-audio features obtained. Namely,

$$\mathbf{m} = \frac{1}{T} \sum_{j=1}^T \Psi(\tilde{\mathbf{z}}_j^I, \tilde{\mathbf{z}}_j^A). \quad (9)$$

$\mathbf{m}$  is a feature summary of the useful moments selected by IMGAUD-SKIMMING. A final fully-connected layer followed by  $\text{Softmax}(\cdot)$  takes  $\mathbf{m}$  as input and makes predictions of action categories. The network is then trained with cross-entropy loss and video-level action label annotations.

While we optimize the IMGAUD-SKIMMING network for a fixed number of  $T$  steps during training, at inference time we can stop early at any step depending on the computation budget. Moreover, instead of using all indexing features, we can also use a subset of indexing features to accelerate inference with the help of feature interpolation. See Sec. 4.2 for details about the efficiency and accuracy trade-off when using sparse indexing features and early stopping.

## 4. Experiments

Using a total of 4 datasets, we evaluate our approach for accurate and efficient clip-level action recognition (Sec. 4.1) and video-level action recognition (Sec. 4.2).

**Datasets:** Our distillation network is trained on Kinetics [33], and we evaluate on four other datasets: Kinetics-Sounds [5], UCF-101 [56], ActivityNet [9], and Mini-Sports1M [32]. Kinetics-Sounds and UCF-101 contain only short trimmed videos, so we only test on them for clip-level recognition; ActivityNet contains videos of various lengths, so it is used as our main testbed for both clip-level and video-level recognition; Mini-Sports1M contains only long untrimmed videos, and we use it for evaluation of video-level recognition. See Supp. for details of these datasets.

**Implementation Details:** We implement in PyTorch. For IMGAUD2VID, the R(2+1)D-18 [62] teacher model takes 16 frames of size  $112 \times 112$  as input. The student model uses a ResNet-18 network for both the visual and audio streams, which take the starting RGB frame of size  $112 \times 112$  and a 1-channel audio-spectrogram of size  $101 \times 40$  (1 sec. audio segment) as input, respectively. We use  $\lambda = 100$  for the distillation loss in Equation 4. For IMGAUD-SKIMMING, we use a one-layer LSTM with 1,024 hidden units and a dimension of 512 for the indexing key vector. We use  $T = 10$  time steps during training. See Supp. for details.

#### 4.1. Clip-level Action Recognition

First, we directly evaluate the performance of the image-audio network distilled from the video model. We fine-tune on each of the three datasets for clip-level recognition and compare against the following baselines:

- **Clip-based Model:** The R(2+1)D-18 teacher model.
- **Image-based Model (distilled/undistilled):** A ResNet-18 frame-based model. The undistilled model is pre-trained on ImageNet, and the distilled model is similar to our method except that the distillation is performed using only the visual stream.
- **Audio-based Model (distilled/undistilled):** The same as the image-based model except here we only use the audio stream for recognition and distillation. The model is pre-trained on ImageNet to accelerate convergence.
- **Image-Audio Model (undistilled):** The same image-audio network as our method but without distillation.

For each baseline, we use the corresponding model as initialization and fine-tune on the same target dataset for clip-based action recognition. Note that our purpose here is not to compete on recognition accuracy using R(2+1)D-18 (or any other expensive video features), but rather to demonstrate our distilled image-audio features can approximate its recognition accuracy much more efficiently.

Figure 5 compares the accuracy vs. efficiency for our approach and the baselines. Our distilled image-audio network achieves accuracy comparable to that of the clip-based teacher model, but at a much reduced computational cost. Moreover, the models based on image-only or audio-only distillation produce lower accuracy. This shows that the image or audio alone is not sufficient to hallucinate the

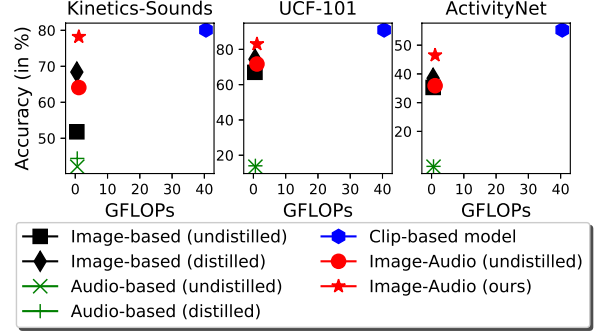


Figure 5: Clip-level action recognition on Kinetics-Sounds, UCF-101, and ActivityNet. We compare the recognition accuracy and the computational cost of our model against a series of baselines. Our IMGAUD2VID approach strikes a favorable balance between accuracy and efficiency.

video descriptor, but when combined they provide sufficiently complementary information to reduce the accuracy gap with the true (expensive) video-clip descriptor.

To understand when audio helps the most, we compute the  $\mathcal{L}_1$  distance of the hallucinated video descriptor to the ground-truth video descriptor by our IMGAUD2VID distillation and the image-based distillation. The top clips for which we best match the ground-truth tend to be dynamic scenes that have informative audio information, *e.g.*, grinding meat, jumpstyle dancing, playing cymbals, playing bagpipes, wrestling, and welding. See Supp. for examples.

#### 4.2. Untrimmed Video Action Recognition

Having demonstrated the clip-level performance of our distilled image-audio network, we now examine the impact of the IMGAUD-SKIMMING module on video-level recognition. We evaluate on ActivityNet [9] and Mini-Sports1M [32], which contain long untrimmed videos.

**Efficiency & accuracy trade-off.** Before showing the results, we introduce how we use feature interpolation to further enhance the efficiency of our system. Apart from using features from all  $N$  time stamps as described in Sec. 3.3, we experiment with using *sparse* indexing features extracted from a subset of image-audio pairs, *i.e.*, subsampling along the time axis. Motivated by the locally-smooth action feature space [13] and based on our empirical observation that neighboring video features can be linearly approximated well, we synthesize the missing image and audio features by computationally cheap linear interpolation to generate the full feature sequences of length  $N$ . Figure 6a shows the recognition results when using different subsampling factors. We can see that recognition remains robust to even aggressive subsampling of the indexing features.

Next we investigate early stopping as an additional means to reduce the computational cost. Instead of repeating the skimming procedure for 10 times as in the training stage, we can choose to stop early after a few recurrent

	RANDOM	UNIFORM	FRONT	CENTER	END	SCSAMPLER [36]	DENSE	LSTM	NON-RECURRENT	Ours (sparse / dense)
ActivityNet	63.7	64.8	39.0	59.0	38.1	69.1	66.3	63.5	67.5	<b>70.3 / 71.1</b>
Mini-Sports1M	35.4	35.6	17.1	29.7	17.4	38.4	37.3	34.1	38.0	<b>39.2 / 39.9</b>

Table 1: Video-level action recognition accuracy (in %) on ActivityNet (# classes: 200) and Mini-Sports1M (# classes: 487). Kinetics-Sounds and UCF-101 consist of only short trimmed videos, so they are not applicable here. Our method consistently outperforms all baseline methods. Ours (sparse) uses only about 1/5 the computation cost of the last four baselines, while achieving large accuracy gains. See Table 2 for more computation cost comparisons.

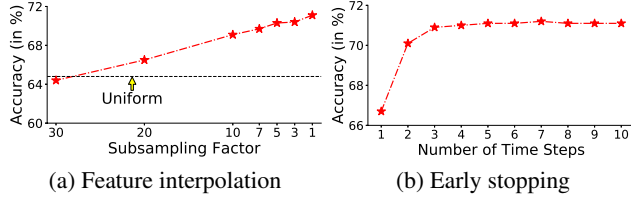


Figure 6: Trade-off between efficiency and accuracy when using sparse indexing features or early stopping on ActivityNet. Uniform denotes the UNIFORM baseline in Table 1.

steps. Figure 6b shows the results when stopping at different time steps. We can see that the first three steps yield sufficient cues for recognition. This suggests that we can stop around the third step with negligible accuracy loss. See Supp. for a similar observation on Mini-Sports1M.

**Results.** We compare our approach to the following baselines and several existing methods [80, 15, 77, 75, 36]:

- **RANDOM:** We randomly sample 10 out of the  $N$  time stamps, and average the predictions of the image-audio pairs from these selected time stamps using the distilled image-audio network.
- **UNIFORM:** The same as the previous baseline except that we perform uniform sampling.
- **FRONT / CENTER / END:** The same as before except that the first / center / last 10 time stamps are used.
- **DENSE:** We average the prediction scores from all  $N$  image-audio pairs as the video-level prediction.
- **SCSAMPLER [36]:** We use the idea of [36] and select the 10 image-audio pairs that yield the largest confidence scores from the image-audio classifier. We average their predictions to produce the video-level prediction.
- **LSTM:** This is a one-layer LSTM as in our model but it is trained and tested using all  $N$  image-audio features as input sequentially to predict the action label from the hidden state of the last time step.
- **NON-RECURRENT:** The same as our method except that we only use a single query operation without the recurrent iterations. We directly obtain the 10 time stamps from the indexes of the 10 largest attention weights.

Table 1 shows the results. Our method outperforms all the baselines. The low accuracy of RANDOM / UNIFORM / FRONT / CENTER / END indicates the importance of the

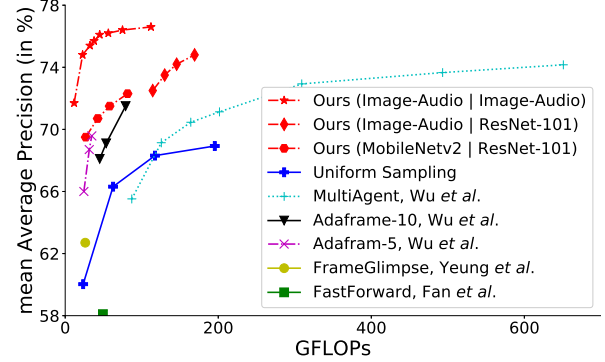


Figure 7: Comparisons with other frame selection methods on ActivityNet. We directly quote the numbers reported in AdaFrame [77] and MultiAgent [75] for all the baseline methods and compare the mAP against the average GFLOPs per test video. See text for details.

context-aware selection of useful moments for action recognition. Using sparse indexing features (with a subsampling factor of 5), our method outperforms DENSE (the status quo of how most current methods obtain video-level predictions) by a large margin using only about 1/5 of its computation cost. Our method is also better and faster than SCSAMPLER [36], despite their advantage of densely evaluating prediction results on all clips. LSTM performs comparably to RANDOM. We suspect that it fails to aggregate the information of all time stamps when the video gets very long. NON-RECURRENT is an ablated version of our method, and it shows that the design of recursive prediction of the “next” interesting moment in our method is essential. Both LSTM and NON-RECURRENT support our contribution as a whole framework, *i.e.*, iterative attention based selection.

#### Comparison to state of the art frame selection methods.

Fig. 7 compares our approach to state-of-the-art frame selection methods given the same computational budget. The results of the baselines are quoted from AdaFrame [77] and MultiAgent [75], where they both evaluate on ActivityNet. For fair comparison, we test a variant of our method with only the visual modality, and we use the same ResNet-101 features for recognition. Our framework also has the flexibility of using cheaper features for indexing (frame selection). See Supp. for details about the single-modality architecture of our IMGAUD-SKIMMING network and how we use different features for indexing and recognition. We



Figure 8: Qualitative examples of 5 uniformly selected moments and the first 5 visually useful moments selected by our method for two videos of the actions *throwing discus* and *rafting* in ActivityNet. The frames selected by our method are more indicative of the corresponding action.

use three different combinations denoted as *Ours* (“*indexing features*” | “*recognition features*”) in Fig. 7, including using MobileNetv2 [51] features for efficient indexing similar to [77]. Moreover, to gauge the impact of our IMGAUD2VID step, we also report the results obtained by using image-audio features for recognition.

Our method consistently outperforms all existing methods and achieves the best balance between speed and accuracy when using the same recognition features, suggesting the accuracy boost can be attributed to our novel differentiable indexing mechanism. Furthermore, with the aid of IMGAUD2VID distillation, we achieve much higher accuracy with much less computation cost; this scheme combines the efficiency of our image-audio clip-level recognition with the speedup and accuracy enabled by our IMGAUD-SKIMMING network for video-level recognition.

**Comparison to the state of the art on ActivityNet.** Having compared our skimming approach to existing methods for frame selection, now we compare to state-of-the-art activity recognition models that forgo frame selection. For fair comparison, we use the ResNet-152 model provided by [75]. This model is pre-trained on ImageNet and fine-tuned on ActivityNet with TSN-style [69] training. As shown in Table 2a, our method consistently outperforms all the previous state-of-the-art methods. To show that the benefits of our method extend even to more powerful but expensive features, we use R(2+1)D-152 features for recognition in Table 2b. When using R(2+1)D-152 features for both indexing and recognition, we outperform the dense approach while being  $10\times$  faster. We can still achieve comparable performance to the dense approach if using our image-audio features for indexing, while being  $20\times$  faster.

### 4.3. Qualitative Analysis

Figure 8 shows frames selected by our method using the visual modality versus those obtained by uniform sampling.

	Backbone	Pre-trained	Accuracy	mAP
IDT [66]	–	ImageNet	64.7	68.7
C3D [61]	–	Sports1M	65.8	67.7
P3D [49]	ResNet-152	ImageNet	75.1	78.9
RRA [88]	ResNet-152	ImageNet	78.8	83.4
MARL [75]	ResNet-152	ImageNet	79.8	83.8
Ours	ResNet-152	ImageNet	<b>80.3</b>	<b>84.2</b>

(a) Comparison to prior work with ResNet-152 features.

	Indexing	Recognition	mAP	TFLOPs
Dense	–	R(2+1)D-152	88.9	25.9
Uniform	–	R(2+1)D-152	87.2	1.26
Ours	Image-Audio	R(2+1)D-152	88.5	1.31
Ours	R(2+1)D-152	R(2+1)D-152	<b>89.9</b>	2.64

(b) Accuracy vs. Efficiency with R(2+1)D-152 features.

Table 2: ActivityNet comparison to SOTA methods.

The frames chosen by our method are much more informative of the action in the video compared to those uniformly sampled. See Supp. video<sup>1</sup> for examples of acoustically useful moments selected by our method.

We can inspect per-class performance to understand what are the classes that benefit the most from our skimming mechanism compared to uniform sampling. The top classes in descending order of accuracy gain are: cleaning sink, beer pong, gargling mouthwash, painting furniture, archery, laying tile, and triple jump—classes where the action is sporadic and is often exhibited over a short segment of the video. See Supp. for more analysis.

## 5. Conclusion

We presented an approach to achieve both accurate and efficient action recognition in long untrimmed videos by leveraging audio as a previewing tool. Our IMGAUD2VID distillation framework replaces the expensive clip-based model by a lightweight image-audio based model, enabling efficient clip-level action recognition. Moreover, we propose an IMGAUD-SKIMMING network that iteratively selects useful image-audio pairs, enabling efficient video-level action recognition. Our work strikes a favorable balance between speed and accuracy, and we achieve state-of-the-art results for video action recognition using few selected frames or clips. In future work, we plan to investigate salient spatial region selection along with our temporal frame selection, which can potentially lead to finer granularity of action understanding with improved efficiency, as well as extensions to allow the multi-label setting.

**Acknowledgements:** Thanks to Bruno Korbar, Zuxuan Wu, and Wenhao Wu for help with experiments and to Weiyao Wang, Du Tran, and the UT Austin vision group for helpful discussions.

<sup>1</sup>[http://vision.cs.utexas.edu/projects/listen\\_to\\_look/](http://vision.cs.utexas.edu/projects/listen_to_look/)



## References

- [1] Cisco visual networking index: Forecast and trends, 2017–2022 white paper.
- [2] T. Afouras, J. S. Chung, and A. Zisserman. The conversation: Deep audio-visual speech enhancement. In *Inter-speech*, 2018.
- [3] S. Albanie, A. Nagrani, A. Vedaldi, and A. Zisserman. Emotion recognition in speech using cross-modal transfer in the wild. In *ACM Multimedia*, 2018.
- [4] H. Alwassel, F. Caba Heilbron, and B. Ghanem. Action search: Spotting actions in videos and its application to temporal action localization. In *ECCV*, 2018.
- [5] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *ICCV*, 2017.
- [6] R. Arandjelović and A. Zisserman. Objects that sound. In *ECCV*, 2018.
- [7] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016.
- [8] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, 2017.
- [9] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.
- [10] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.
- [11] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng. Multi-fiber networks for video recognition. In *ECCV*, 2018.
- [12] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.
- [13] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. Temporal cycle-consistency learning. In *CVPR*, 2019.
- [14] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. In *SIGGRAPH*, 2018.
- [15] H. Fan, Z. Xu, L. Zhu, C. Yan, J. Ge, and Y. Yang. Watching a small portion could be as good as watching all: Towards efficient video classification. In *IJCAI*, 2018.
- [16] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *ICCV*, 2019.
- [17] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.
- [18] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.
- [19] C. Gan, H. Zhao, P. Chen, D. Cox, and A. Torralba. Self-supervised moving vehicle tracking with stereo sound. In *ICCV*, 2019.
- [20] R. Gao, R. Feris, and K. Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018.
- [21] R. Gao and K. Grauman. 2.5d visual sound. In *CVPR*, 2019.
- [22] R. Gao and K. Grauman. Co-separating sounds of visual objects. In *ICCV*, 2019.
- [23] W. W. Gaver. What in the world do we hear?: An ecological approach to auditory event perception. *Ecological psychology*, 1993.
- [24] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell. ActionVLAD: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017.
- [25] R. Girdhar, D. Tran, L. Torresani, and D. Ramanan. Distinit: Learning video representations without a single labeled video. In *ICCV*, 2019.
- [26] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *NeurIPS*, 2014.
- [27] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.
- [28] S. Gupta, J. Hoffman, and J. Malik. Cross modal distillation for supervision transfer. In *CVPR*, 2016.
- [29] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [30] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis. Representing videos using mid-level discriminative patches. In *CVPR*, 2013.
- [31] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. In *CVPR*, 2014.
- [32] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.
- [33] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.
- [34] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen. Epicfusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019.
- [35] B. Korbar, D. Tran, and L. Torresani. Co-training of audio and video representations from self-supervised temporal synchronization. In *NeurIPS*, 2018.
- [36] B. Korbar, D. Tran, and L. Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *ICCV*, 2019.
- [37] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.
- [38] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012.
- [39] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019.
- [40] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018.
- [41] X. Long, C. Gan, G. De Melo, J. Wu, X. Liu, and S. Wen. Attention clusters: Purely attention based local feature integration for video classification. In *CVPR*, 2018.

- [42] B. Mahasseni, M. Lam, and S. Todorovic. Unsupervised video summarization with adversarial lstm networks. In *CVPR*, 2017.
- [43] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *NeurIPS*, 2014.
- [44] P. Morgado, N. Vasconcelos, T. Langlois, and O. Wang. Self-supervised generation of spatial audio for 360° video. In *NeurIPS*, 2018.
- [45] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018.
- [46] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *CVPR*, 2016.
- [47] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016.
- [48] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, 2014.
- [49] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.
- [50] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012.
- [51] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.
- [52] A. Senocak, T.-H. Oh, J. Kim, M. Yang, and I. S. Kweon. Learning to localize sound sources in visual scenes: Analysis and applications. *TPAMI*, 2019.
- [53] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017.
- [54] Z. Shou, X. Lin, Y. Kalantidis, L. Sevilla-Lara, M. Rohrbach, S.-F. Chang, and Z. Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *CVPR*, 2019.
- [55] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.
- [56] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [57] Y.-C. Su and K. Grauman. Leaving some stones unturned: dynamic feature prioritization for activity detection in streaming video. In *ECCV*, 2016.
- [58] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *NeurIPS*, 2015.
- [59] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019.
- [60] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018.
- [61] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.
- [62] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.
- [63] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *TPAMI*, 2017.
- [64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [65] O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. In *NeurIPS*, 2015.
- [66] H. Wang and C. Schmid. Action recognition with improved trajectories. In *CVPR*, 2013.
- [67] L. Wang, Y. Qiao, and X. Tang. Motionlets: Mid-level 3d parts for human motion recognition. In *CVPR*, 2013.
- [68] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017.
- [69] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.
- [70] W. Wang, D. Tran, and M. Feiszli. What makes training multi-modal networks hard? *arXiv preprint arXiv:1905.12681*, 2019.
- [71] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018.
- [72] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.
- [73] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019.
- [74] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl. Compressed video action recognition. In *CVPR*, 2018.
- [75] W. Wu, D. He, X. Tan, S. Chen, and S. Wen. Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In *ICCV*, 2019.
- [76] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue. Multi-stream multi-class fusion of deep networks for video classification. In *ACM-MM*, 2016.
- [77] Z. Wu, C. Xiong, C.-Y. Ma, R. Socher, and L. S. Davis. Adaframe: Adaptive frame selection for fast video recognition. In *CVPR*, 2019.
- [78] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.
- [79] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017.
- [80] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.
- [81] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.
- [82] K. Zhang, K. Grauman, and F. Sha. Retrospective encoders for video summarization. In *ECCV*, 2018.
- [83] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba. The sound of pixels. In *ECCV*, 2018.
- [84] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017.

- [85] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.
- [86] H. Zhou, Z. Liu, X. Xu, P. Luo, and X. Wang. Vision-infused deep audio inpainting. In *ICCV*, 2019.
- [87] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg. Visual to sound: Generating natural sound for videos in the wild. In *CVPR*, 2018.
- [88] C. Zhu, X. Tan, F. Zhou, X. Liu, K. Yue, E. Ding, and Y. Ma. Fine-grained video categorization with redundancy reduction attention. In *ECCV*, 2018.
- [89] M. Zolfaghari, K. Singh, and T. Brox. Eco: Efficient convolutional network for online video understanding. In *ECCV*, 2018.