

Master Thesis

Egocentric Action Recognition from Noisy Videos

(ノイズを含む一人称視点映像からの動作認識)

Lijin Yang

Advisor: Professor Yoichi Sato

Submission Date: Jan 30th, 2020



Department of Information and Communication Engineering
Graduate School of Information Science and Technology
The University of Tokyo

Advisor

Prof. Yoichi Sato

Abstract

Recognizing human actions in videos can make machines better interpret the behavior of human beings. As one of the basic fields of computer vision, action recognition has attracted significant research attention. In particular, with the recent advance of deep learning, there is a huge boost of action recognition performance. Deep learning models that takes as input video clips can reliably recognize the actions compared with traditional methods using hand-crafted features.

However, one of the main drawbacks of existing deep learning models is that the input video clips are assumed to be pre-processed. This pre-processing step includes trimming unrelated parts and deleting ambiguous noisy parts of the video clip. Other than the huge human labour that is needed in the pre-processing, the assumption that the videos are clean (pre-processed, noise free) is not realistic and strongly restricts the real world application of action recognition models. In real world videos, actions are performed naturally without restriction. This would result in the video clips contain lots of un-informative or noisy frames. If we directly using the noisy video clips as input to the existing methods, there would be a significant performance decrease. We argue that an automatic frame selection method is needed for more robust action recognition in real world applications.

In this work, we aim at alleviating the pre-processing problem and make action recognition methods more robust under the input of un-processed videos. To best leverage the existing success of action recognition models, we propose a plug-and-play module for automatically pre-process the input video. The proposed module acts as a frame selection tool and thus could be applied on top of any existing models for action recognition. With the proposed module, the input to the recognition system could directly be un-processed videos. The un-processed videos are first selected by the proposed module. The selected informative frames are then used as input to the recognition module. By this means we can allow action recognition models to take as input noisy video clips without pre-processing, while in the mean time robustly output reliable action recognition results.

Technically speaking, the proposed module contains two major sub-modules: the sampler network and the evaluator network. The sampler network samples important frames from long unprocessed video clips for action recognition modules. The evaluator acts like a teacher to the sampler network by providing feedback on how well the output of sampler is. By jointly end-to-end training the sampler and the evaluator using the final action recognition loss, we could get a sampler that could

select useful frames best for action recognition. In the inference stage, the evaluator is discarded thus further accelerates the inference even with long video clip as input.

We conduct our method on the public EGTEA dataset. EGTEA dataset contains egocentric videos taken from head mounted cameras. More than 30 participants were asked to cook in a kitchen with a certain recipe. As most participants are not experts in cooking, there are many noisy frames in most of the actions. Thus we use this dataset to validate the effectiveness of our method. Experiments show that our proposed sampler can successfully select better frames for action recognition, and that with the evaluator the sampler could be better trained to even boost the selection performance.

Contents

List of Figures	1
List of Tables	3
1 Introduction	5
1.1 Overview	5
1.2 Thesis Outlines	7
2 Related Work	9
2.1 Action Recognition	9
2.1.1 Traditional Methods Using Hand-crafted Features	9
2.1.2 CNN based Deep Learning Methods	10
2.1.3 RNN based Deep Learning Methods	11
2.1.4 Hybrid Models in the Context of Deep Learning	11
2.2 Video Frame Selection	12
2.2.1 Video Summarization	12
2.2.2 Video Highlight Detection	13
2.2.3 Action Localization	13
2.2.4 Frame Selection for Efficiency	14
3 Proposed Method	17
3.1 Model Architecture	17
3.2 Recognition Module	17
3.3 Sampler Module	18
3.4 Evaluator Module	20
3.5 Training and Implementation Details	21
3.5.1 Training	21
3.5.2 Implementation details	23
4 Experiments	25
4.1 Dataset	25
4.2 Experimental Setup	25
4.2.1 Recognizing interrupted actions	25
4.2.2 Localizing informative frames in super noisy videos	27
4.3 Recognizing Interrupted Actions	28
4.3.1 Comparison with baseline methods	28
4.3.2 Cooperating with multiple recognition backbones	29

4.3.3	Visualizing frame selection	31
4.4	Localizing Informative Frames in Super Noisy Videos	31
4.5	Ablation Study	34
4.5.1	Results on recognizing interrupted actions	35
4.5.2	Results on localizing useful frames from super noisy videos	35
5	Discussion	37
5.1	Impact of Different Kinds of Loss in Sampler Adaption	37
5.2	Comparison between Different Attention Mechanisms	38
5.3	Limitation of Our Model	39
6	Conclusion and Future work	41
	Acknowledgments	43
	References	45
	List of Publications	53

List of Figures

1.1	Selecting frames for action recognition	6
3.1	Architecture of our Proposed Model	18
4.1	Example images in EGTEA dataset	26
4.2	Example of none-action frames.	27
4.3	Visualization of frame selection results for recognizing interrupted actions.	31
4.4	Quantitative results of selected frames noise rate	32
4.5	Visualization of frames selection results on super noisy videos.	33

List of Tables

- 4.1 Action recognition performance comparison with baseline methods. . . 28
- 4.2 Quantitative comparison on cooperating with different backbones. . . 30
- 4.3 Ablation study for different parts of our model on recognizing interrupted actions. 35
- 4.4 Ablation study for different parts of our model on localizing useful frames in super noisy videos. 36
- 5.1 Quantitative comparison on using different attention mechanisms. . . 38

1 Introduction

1.1 Overview

There has been increasingly amount of attention from the research community on video-based action recognition [WS13, GYZ⁺16]. This is because of action recognition has wide applications in many areas such as security and human behavior analysis. With recent advance of deep learning techniques, action recognition can be more accurately performed by applying a deep CNN over fixed-length video clips [SZ14, FPZ16]. Most modern action recognition models operate on manually selected video clips in which irrelevant or noisy parts are filtered out. However, assuming input clips are pre-processed has great limitation in real world applications and is also very unrealistic especially in first-person (or egocentric) videos, where camera wearers may often be interrupted during the middle of an action. Moreover, the manual pre-processing of videos tend to be extremely time-consuming, this is especially important when the amount of data becomes large.

In this work we aim at alleviating the limitation of existing action recognition models by introducing a plug-and-play model on top. The model can take as input unprocessed video clips and output informative frame/clips for better action recognition. By introducing a new Sampler-Evaluator model on top of any existing action recognition models, we can make the best use of the success of recent action recognition methods (for example [WGGH18, SEL19, WG18]) without the need of modifying or tuning them.

Technically speaking, we propose a Sampler-Evaluator scheme for learning to select informative frames for better action recognition. There are two main components of the proposed model: sampler and evaluator. The sampler network takes as input long and noisy unprocessed video clips, and outputs a sequence of selected informative frames. Since the ground-truth of noisy frames are unknown, we use an evaluator network for better training the sampler. The evaluator network evaluates the quality of the selected sequence, using the information that how well are the selected sequence recognized by existing action recognition models. The role of evaluator acts like a teacher that provides feedback for better training the sampler network. During the inference step, the evaluator is discarded. We show in experiments that by adding the sampler we can better recognize action from noisy video inputs, and that by adding the evaluator in training it is able to boost the final action recognition performance even more. In the experiments we also demon-

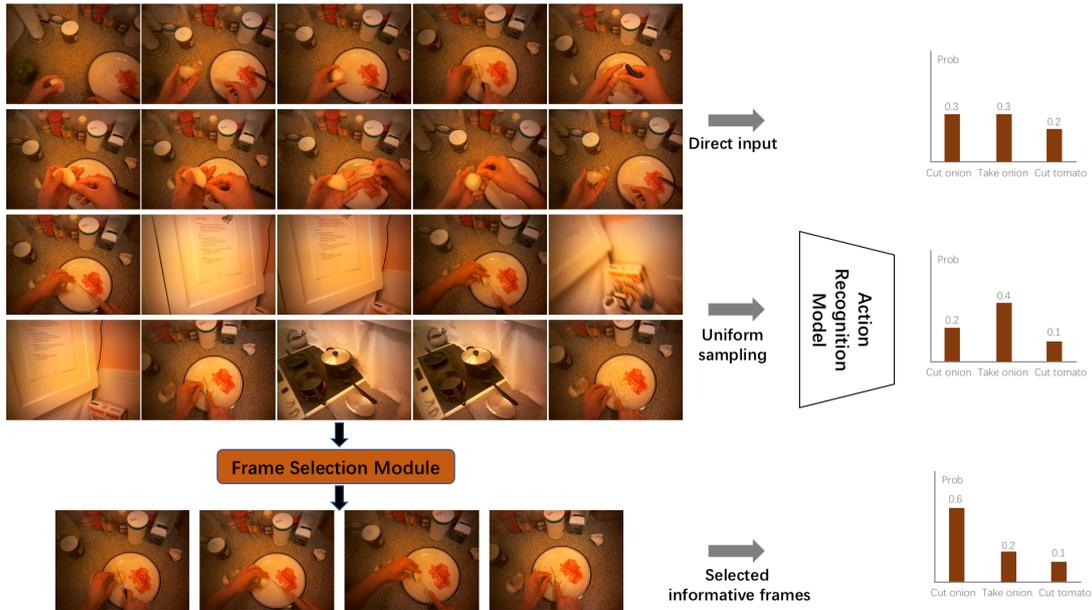


Figure 1.1: Selecting frames for action recognition. Given an unprocessed video clip containing a “cut onion” action, we showcase three sampling strategies for action recognition. (1) Direct input all frames: (2) Uniform sampling and (3) select informative frames. Since during the cutting action, the camera wearer peels the onion with his hand, and also looked around several times, either direct input all frames or uniform sample from the clip cannot make the action recognition model output a correct result. In noisy video input like this, it is essential that we select informative frames for better action recognition.

strate that the sampler can effectively filter out different types of noise and select informative frames/clips for better action recognition.

In summary, our main contributions are three-folded as follows:

- Firstly, we propose a novel model for selecting frames from noisy video inputs for better action recognition. Our model can be built on top of any existing models for action recognition and is end-to-end trainable without the need of frame selection ground-truth. To the best of our knowledge, this is the first end-to-end trainable frame-selection model for improving action recognition performance, using only action recognition as supervision.
- Secondly, propose a novel sampler-evaluator mechanism for enabling end-to-end training the proposed model. By using the evaluator that could measure the performance of the sampler network and give the sampler feedback, our model could be end-to-end trained without using ground-truth frame selections which is too costly to acquire in real world data.
- Thirdly, we demonstrate in experiments that the proposed model can successfully filter out noisy components in the video input and increase the action

recognition performance by selecting informative frames.

1.2 Thesis Outlines

The rest of this thesis is organized as follows. In Chapter 2, we first provide an overview of recent related works on action recognition and video based frame selection. For each field of related works we select at least three closely related methods to be described in detail. We then present our method in Chapter 3, our method is a plug-and-play model on top of any existing action recognition models. We will introduce the sampler and evaluator as the main component of our proposed model. We also introduce the training strategy. In Chapter 4, we evaluate our method and show its superiority over other baseline methods. We conduct extensive experiments under multiple noise conditions to validate the effectiveness of our method. Another reason why we use multiple noise conditions is that we would like to simulate the realistic noise that could happen during the video recording or video transmission. Current limitations are also presented and possible solutions and other modifications are discussed. Finally, Chapter 5 summarizes this thesis.

2 Related Work

The goal of our work is to achieve better action recognition under very noisy inputs. Thus in this section we review related works from mainly two perspectives: action recognition [PWWQ16, GYZ⁺16, WQT, CZ17, DFS⁺18, FPW16, FPZ16, XSH⁺18] and frame selection [KTT19, GCGS14, GGVG15, MMJ⁺18, MJM⁺18, CHCNG16, JVGJ⁺14, KL14, PBTM17]. In the following of this section we will introduce the overall research of action recognition and frame selection, together with some more related works explained in detail.

2.1 Action Recognition

As one of the fundamental research topic in computer vision, action recognition from videos has been extensively studied in recent years [ZSB18, YXL18, XSH⁺18, WMD18, CKL⁺18, ZSZZ18, ZAOT18]. Basically, most works try to model the spatial and temporal information in a video for action recognition. Earlier works design hand crafted features for describing the object and motion in the video so as to do the recognition. With the help of deep learning techniques, the performance of methods using deep neural networks have surpassed those who use hand crafted features.

2.1.1 Traditional Methods Using Hand-crafted Features

Before the age of deep learning, many representative works [WS13, JXYY12, WQT14, CWPQ14, NMY15] in action recognition encode motion information in hand crafted features. For example, Wang *et al* [WKSCL11] propose to use dense trajectories. Feature keypoints are sampled densely from each frame and then tracked based on the displacement information from a dense optical flow field. This work is further extended as improved dense trajectory [WS13] that takes into account the camera motion. By compensating the camera motion, the motion inside the video could be better represented, thus better action recognition could be achieved. Other than just using hand-crafted feature alone, some work [WQT15, CLS15] also hand-craft features extracted from deep convolutional neural networks.

Many works also devote to model the temporal structure of action [FGO⁺15, NCF10, WQT13, SJYS15]. Gaidon *et al.* [GHS13] propose to represent each atomic action

seperatedly, they annotated each atomic action for each video and proposed Action Sequence Model for action detection. Niebles *et al.* [NCFF10] proposed to decompose actions into multiple sub-actions, and use latent variables to model the temporal decomposition. For the classification they resorted to the Latent SVM [FGMR09] to learn the model parameters in an iterative approach. Wang *et al* [WQT13] extended the temporal decomposition of complex action into a hierarchical manner using Latent Hierarchical Model and Segmental Grammar Model, respectively. Fernando *et al* [FGO⁺15] modeled the temporal evolution of Bag of Visual Words representations for action recognition.

2.1.2 CNN based Deep Learning Methods

With the rise of deep learning, many works try to design effective models for action recognition using convolutional neural networks (CNNs) [SZ14, SJYS15, SKS⁺18, TBF⁺15, VLS17, WLLVG18, WXW⁺16, WGGH18, WG18]. In action recognition, there are two significant while complementary aspects: appearances and motion. As earlier CNNs only aim at extracting features spatially while ignoring temporal information, Simonyan *et al.* designed a two-stream CNN that takes as input both images and optical flows. This architecture is proved to be effective in capturing both object and motion information. Afterwards, a bunch of works follow their steps to use two stream architecture for action recognition.

For example, Wang *et al.* [WXW⁺16] proposed the Temporal Segment Network (TSN) for modeling long-range temporal structure. Aiming at extracting and utilizing relevant appearance and motion information from long range temporal structure, they build their TSN on top of the two-stream architecture [SZ14]. Since in most videos consecutive frames are usually redundant, instead of densely sampling frames from videos, TSN uses a sparse sampling technique. They first uniformly divide the whole video into several segments, and randomly extract a short video snippet from each of the segment. By late fusion of scores predicted from each of the segments, a better recognition performance is achieved. However, one obvious drawback is the randomness within the sampling process: some critical part of the video may be lost during the random sampling in each video segment.

As an alternative way for capturing motion information in CNNs, 3D CNNs are also extensively studied in the field of action recognition. C3D network [TBF⁺15] firstly extended 2D convolution filters into 3D. By this means the motion information within the videos can be implicitly extracted by the 3D convolution operations. To reduce the computational cost of 3D CNNs, some works [QYM17, XSH⁺18] used two 1D convolutional kernels for replacement of 3D convolutional kernels. In 2017, Carreira *et al.* [CZ17] extended the BN-Inception network [SVI⁺16] to 3D convolution network named I3D. They do this by inflating the 2D convolutional kernels into 3D, thus enabling the network to extract seamless spatio-temporal feature from videos while leveraging the Imagnet architecture designs. Following works

designed 3D-Resnet [WGGH18] by inflating the 2D kernels in ResNet [HZRS16] into 3D kernels. Remarkable performance gain is achieved. Most recent state-of-the-art action recognition models [WG18, CKL⁺18, LLL19b, KPvD⁺19] are based on the 3D-Resnet as backbone.

2.1.3 RNN based Deep Learning Methods

Besides using optical flow as input or using 3D convolutional neural networks, another direction to capture motion information in videos is to use Recurrent Neural Networks (RNN). One of the most powerful recurrent neural network is Long Short Term Memory (LSTM). The main idea of LSTM architecture is its memory cell, input gate, output gate, and forget gate, which can encode the temporal order of features over time. Its non-linear gating units can regulate the information to flow into or flow out of the memory cell [GSK⁺16, UAM⁺17]. For example, Singh *et al.* [SMP⁺17] compared the performance of action recognition using LSTM, Naive Bayes [TIL04], Hidden Markov Model (HMM) [DBPV05], and Conditional Random Fields (CRF) [VKNEK08]. Experiments on public datasets demonstrate the effectiveness of LSTM, as it outperforms all other compared baselines. However, using RNNs alone cannot perform good results on action recognition. Researchers often combine RNN with features extracted by CNN for better recognition accuracy.

2.1.4 Hybrid Models in the Context of Deep Learning

CNNs excel at extracting powerful representations from images while RNNs are good at modeling temporal evolution information. Based on their unique strengths, several researches focus on combining the strength of CNN and RNN by using features extracted from CNNs as input to RNN [ZGY⁺18, SEL19, SL18, ZXL⁺18].

A recent work proposed by Sudhakaran *et al.* leveraged LSTM and applied attention mechanism for better emphasizing important regions in the image. They present Long Short-Term Attention (LSTA) which is a modified version of LSTM that addresses shortcomings of LSTM: LSTMs do not have spatial attention to focus on more critical part of images. When the discriminative information in the input sequence can be spatially localized, conventional LSTM cannot emphasize this spatial region. With this in mind, adding attention mechanisms to LSTM cells is a natural solution. Attention mechanism was proposed for focusing attention on features that are relevant for the task to be recognized. For generating the spatial attention, most existing techniques [SL18, PFR17] consider each frame independently. Since video frame sequences have an absolute temporal consistency, per frame processing results in the loss of valuable information. In this work, the authors propose LSTA that could learn spatial attention map in a top-down fashion utilizing prior information encoded in one CNN pre-trained for object recognition and another pre-trained for action recognition.

The proposed LSTA extends LSTM with two novel components: recurrent attention and output pooling. The recurrent attention part tracks a weight map to focus on relevant features, while the output pooling component introduces a high-capacity output gate. At the core of both is a pooling operation that selects one out of a pool of specialized mappings to realize smooth attention tracking and flexible output gating. The effectiveness of LSTA is evaluated on GTEA [LYR15] and EGTEA [LLR18] datasets. LSTA is now the state-of-the-art method for egocentric action recognition. In this work, we also explore the feasibility to combine CNN with RNN for better model performance.

2.2 Video Frame Selection

Another research topic closely related with our work is video frame selection. The task of video frame selection have various applications, ranging from video summarization [RYW18, XE19], action localization [KWFS17, CVS⁺18, ZHT⁺19, GGCN19, LLL⁺19a] to efficient video processing [KTT19]. Our work takes a step towards better action recognition in noisy data, thus selecting a set of robust and reasonable is a part of our goal. In the following subsections, we introduce some related works about different aspects of frame selection that inspired our model design.

2.2.1 Video Summarization

With the world entering the digital era, large amount of videos appear on the internet. Video data are often redundant: it is tiresome for human to observe all of the video data for extracting useful information within. The technique of video summarization, whose goal is to select a subset of the frames to create a summary video that optimally captures the important information of the input video [ZCSG16, MLT17, RYW18], is very important for the tasks such as video search and browsing.

Recently, fully convolutional network (FCN) based video summarization has been proposed [RYW18] and is proved to outperform RNN based video summarization techniques [ZCSG16, MLT17]. In the paper of Rochan *et al.*, temporal FCN is used to capture long range temporal information by processing all frames simultaneously using the convolution operation. Based on the intuition that important frames should be visually diverse, their model could also perform unsupervised video summarization by encouraging the diversity of the selected frames.

To validate the performance of FCN on video summarization, experiments are done on two public datasets: SumMe [GGRVG14] and TVSum [SVSJ15]. Their method outperforms other models by a large margin, indicating the usefulness of FCN on video summarization. More importantly, the authors tested video summarization performance in a "Transfer Setting", where the model is trained on one dataset and

tested on another dataset. This is a challenging setting since the data distribution may differ greatly between datasets. Experiments show that although in the standard setting, FCN performs better than other methods (such as RNN based method [ZCSG16]), in the transfer setting the FCN model and the RNN models perform comparably. This indicates that current state-of-the-art works is still hard in generalization to unseen data.

2.2.2 Video Highlight Detection

Video highlight detection is another important application of video frame selection, especially in sports videos. Due to the difficulty in getting ground truth of video highlight, Yao *et al.* [YMR16] propose a deep ranking technique for highlight detection in paired videos. Different than the traditional supervised scheme like [ZCSG16], where the absolute ground truth is given for training a video summarization model, this work determines whether a video segment is highlight or not by comparing it with other video segments. This greatly alleviates the effort of human labeling, and could also get more reliable annotations since relative highlight difference is usually similar even across different human annotators. As one of the contributions of this work, Yao *et al.* collected a new dataset of first person sports videos, and crowd-sourced the annotations.

As another contribution of this work, the deep ranking model takes as input two segments of video. After processing the segments using a Siamese network where two identical branches share the same parameters, the output of each branch is compared using a pairwise ranking loss. This loss encourages the output of the branch which takes input the segment with higher highlight ground truth score to be higher than the other branch. By this scheme, the model could be trained to output a highlight score of a video with only training data with pairwise label. Experiments on the newly collected dataset also validates the effectiveness of their proposed method. In this paper, since we do not have a ground truth of the ranking score, we do not use the ranking scheme for selecting important frames.

2.2.3 Action Localization

Action localization is a task of selecting frames of a certain action out of a video containing both the action and background. This is an emerging field of research [PBTM17, ZHT⁺19] as it benefits other research fields like action recognition and video understanding. Previous approaches on this task can be grouped into two categories: (1) two stage approaches where action proposals are first generated and then classification and boundary refinement is applied on each proposals [SWC16, ZHT⁺19]; and (2) methods with end-to-end architectures integrating the proposal generation and classification in a one-stage framework [BEG⁺17]. Overall, methods in the first category results in better performances while methods in the second

category are faster and easier to train. As an example, Pei *et al.* proposed to use a modified version of LSTM named Temporal Attention-Gated Model (TAGM) for action localization. The TAGM can calculate a saliency score for each input frame in the whole video. We use TAGM as an option of attention module in our proposed method.

Recent researches not only focus on utilizing the spatiotemporal information of the video, but also consider the relation within the video. For example, action proposal relations in the video could help with the action localization task [ZHT⁺19]. Zeng *et al.* [ZHT⁺19] explicitly leveraged the relation between multiple action proposals to improve the action localization result. They make use of the Graph Convolution Network [KW16] for modeling the relations among proposals. In this work we also considered using graph convolution networks for selecting relevant frames from noisy video input, however the result is not satisfactory. We suspect this is because that graph convolution networks are good at capturing global video information but tend to ignore local importance, due to the node-wise average pooling operation [KW16, WG18].

2.2.4 Frame Selection for Efficiency

Frame selection could also improve the efficiency of action recognition algorithms. To increase the recognition accuracy, action recognition models tend to be complex [KTT19], which is an obstacle for the adaptation on mobile devices. Also, videos often contain redundant frames, and standard pooling operations can result in poor video-level recognition, as informative clip features are outnumbered by uninformative features over long unimportant frames.

Very recently, researchers from Facebook AI [KTT19] propose to use a lightweight network to determine the saliency of frames in a long video clip, so as to select frames for the next computationally expensive action recognition step. The core idea is to design an extremely lightweight network (sampler network) that can take as input large number of video frames while in the meantime output a reasonable saliency score. Their proposed way of making the sampler network lightweight is to make use of the compressed image and audio information. While usually CNNs takes as input RGB images, they design their network to take as input one RGB frame followed by 11 motion displacement images. The motion displacement images have only 2 channels, and are resized to 1/16 of the original image size. By this means, the computational cost of processing the input is drastically reduced, while information within the frames are mostly preserved. The sampler network also leverages the MEL-spectrograms from audio segments, since audio information is lightweight but can provide some useful information indicating whether action is happening.

Due to the lack of ground truth saliency scores, the training of the sampler network is not straightforward. As for the image input, the authors simply train the sampler as an action classifier, and the saliency score is computed as the maximum score

over all the classes. As for the audio input, the authors further used the previous classification score and train the audio-based sampler as a saliency ranker. Ranking loss same with [YMR16] which is described in the previous subsection, is used for training the audio-based sampler. The authors evaluated the proposed method on Sports-1M dataset [KTS⁺14], where videos have average length of 5 minutes which is good for validating the method.

Inspired by this work [KTT19], in this work our method also uses action recognition as supervision for training the sampler network. However, different from their work, we add an attention module for acquiring the importance score of each frame. For better training the sampler network, in this work we also add an evaluator network to encourage the sampler to draw better samples for action recognition. Details of our network is described in the next section.

3 Proposed Method

In this section, we will first give an overview of our full model for selecting informative frames from input videos. Then we will go into details about the three major components of our model, especially the Sampler-Evaluator model which could work with any existing action recognition models. In the end, we will introduce the implementation details and the iterative training scheme for training our model.

3.1 Model Architecture

In this work, we aim at selecting informative frames from noisy input videos for better action recognition. Our model can be viewed as the combination of the Action Recognition Module and the plug and play Sampler-Evaluator Model. Figure 3.1 depicts the overall architecture of our model.

Given fixed-length clips with spatial resolution $H \times W$, the Sampler module extracts deep features from a single RGB image at each time step $t \in \{1, \dots, N\}$. Attention scores for each frame are then generated by applying self-attention on the deep features. And the top n frames are selected according to the attention scores. Clip V_s with n selected frames are further used as input for Recognition module to produce the possibilities for each class $\{1, \dots, C\}$. At the same time, the Evaluator module estimates an effectiveness score based on deep features and selected frames generated from the Sampler module. The easier the selected frames are for the Recognition module to make accurate recognition, the closer the score will be to 1. More details of each module is explained in the following sections.

3.2 Recognition Module

The focus of our work is to propose a plug and play frame selection module which can work with any existing action recognition models, so we are not committed to optimizing the existing models, but pay attention to the cooperation between our Sampler-Evaluator modules and the existing models.

As mentioned in Korbar *et al.* [KTT19], the majority of modern action recognition models constrain the number of input frame n to keep memory consumption manageable in both training stage and testing stage. Usually, n frames only allow their

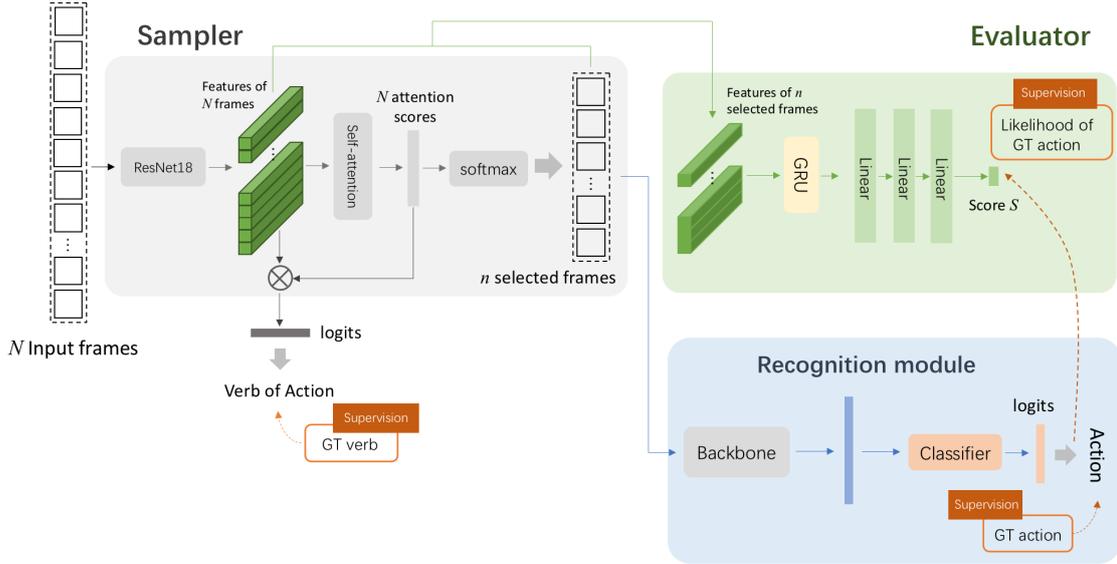


Figure 3.1: Architecture of our Proposed Model. Our proposed model contains three main modules: (1) the Sampler module samples discriminative frames from noisy input frames, (2) the Evaluator module evaluates the performance of the Sampler and provides feedback for better training the Sampler network. (3) The Recognition module which essentially could be any existing models for action recognition.

models to see a handful of seconds at one time, which imposes a horizon for the model to understand the whole input clip. And this problem is particularly salient when the action recognition model goes more complex.

Given n input frames, Recognition module will first encode spatial-temporal information using CNNs or RNNs to deep feature X_R . With regard to 2D networks, $X_R \in \mathbb{R}^{n \times d}$ is extracted from each frame and aggregated on temporal dimension. And for 3D networks, $X_R \in \mathbb{R}^d$ is directly generated for the whole clip.

After extracting deep feature X_R , a fully connected layer is used to produce action classification possibilities \hat{P}_a over all action classes.

$$\hat{P}_a = \text{softmax}(\mathbf{W}X_R + \mathbf{b}) \quad (3.1)$$

3.3 Sampler Module

Our technical interests lie in selecting informative frames which benefit Recognition module to make correct action recognition, so the Sampler module is critical and indispensable in our proposed model. Given N frames, our Sampler module will

select the most informative n frames from them. Obviously, this not only requires the Sampler module to be able to handle more input frames, but also requires it to be salience-sensitive and noise-robust.

Considering computational efficiency, we choose to use down-scaled gray-scale frames as the input of our Sampler module. First of all, we randomly take N gray-scale frames from each long input video clip to form the Sampler input $V \in \mathbb{R}^{N \times H \times W}$. For the efficiency of feature extraction process together with the effectiveness of extracted features, ResNet18 [HZRS16] pretrained on ImageNet [DDS⁺09] is used to generate deep feature $X_f \in \mathbb{R}^{N \times d}$. Afterwards, attention scores $a \in \mathbb{R}^N$ among all N frames are produced by applying attention mechanism on deep feature X_f . High attention score indicates that more relative to other frames, and vice versa.

Attention mechanism in deep neural networks [WJQ⁺17], has been widely exploited in many aspects of image and video processing. Spatial attention endows models with great discrimination, and allow them to liberate from vast information and concentrate on key regions in images.

However, rather than spatial attention, we put more emphasis on introducing temporal attention over N frames to highlight informative frames as well as ignore irrelevant frames. In this work, we exploit the following two methods to calculate attention scores and further generate improved deep features.

Self-attention [VSP⁺17] method was proposed by Vaswani *et al.* in 2017. Different from other methods that take the advantage of CNN or RNN, they employed a transformer architecture to generate scaled dot-product attention, which accelerates computational speed and prompts to capture long-term dependency.

In our work, deep features X_f are first mapped to value embedding $V \in \mathbb{R}^{N \times d_v}$ query embedding $Q \in \mathbb{R}^{N \times d_q}$ and key embedding $K \in \mathbb{R}^{N \times d_k}$. Then similarity between Q and K are calculated by taking dot product of query embedding with key embedding. And the attention scores are generated by applying softmax on scaled similarity.

$$a_t = \text{softmax}\left(\frac{Q_t K_t^T}{\sqrt{d_k}}\right) \quad (3.2)$$

Finally, we can use attention scores to weight the deep features:

$$X_s = \sum_{t=1}^N a_t V_t \quad (3.3)$$

Temporal attention-gated model [PBTM17] Unlike self-attention method, they use attention gate to determine whether ‘keep’ or ‘neglect’ information from current frame to learn informative representation recurrently. Based on this temporal attention gate, they can prize salient clips over redundant or noisy frames.

Following their method, we use bi-directional GRU(Gated Recurrent Unit) to get hidden state \vec{h}_t . By integrating information from both directions, attention score a are inferred as:

$$a_t = \sigma(\mathbf{W}\vec{h}_t + b) \quad (3.4)$$

At each time step t , the hidden state \mathbf{h}_t is updated using previous hidden \mathbf{h}_{t-1} and candidate hidden state \mathbf{h}' . This renewing process is effected by attention scores estimated above. If the attention score of frame t is closer to 1, then a new hidden state will be mainly incorporated from the current input frame. Otherwise, the hidden state will remain similar to the previous one.

$$\mathbf{h}_t = (1 - a_t) \cdot \mathbf{h}_{t-1} + a_t \cdot \mathbf{h}' \quad (3.5)$$

And the candidate hidden state is formulated as the sum of linear transformed \mathbf{h}' and current feature \mathbf{x}_t .

$$\mathbf{h}' = \sigma(\mathbf{W} \cdot \mathbf{h}_{t-1} + \mathbf{U} \cdot \mathbf{x}_t + \mathbf{b}) \quad (3.6)$$

\mathbf{W} and \mathbf{U} refer to liner transformation, and σ denotes the activation function. After finishing recurrent adaptation, we use the last hidden h_N state as an improved deep feature X_s .

We initialize the Sampler module using an action verb recognition task, therefore we use a fully connected layer to get possibilities \hat{P}_v for each verb class.

$$\hat{P}_v = \text{softmax}(\mathbf{W}X_s + \mathbf{b}) \quad (3.7)$$

After initialization, we adopt softmax to preserve the gradient while obtaining the discrete selected frame index I . More details about comparison between two attention mechanisms can be found in Section 5.2.

3.4 Evaluator Module

Since the input of Recognition Module are the indices of selected frames rather than deep features generated by the Sampler module, gradient between Recognition Module and Sampler module cannot be back propagated. In order to enable gradient back propagation, the Evaluator module is indispensable for directing the Sampler module to select frames that benefits Recognition Module to make correct recognition.

To achieve a balance between efficiency and effectiveness, we proposed to use 1-layer GRU(Gated Recurrent Unit) and 3-layer MLP(Multi-Layer Perceptron) as the prototype of Evaluator.

Given deep features $X_s \in \mathbb{R}^{N \times d}$ and frame selection index $I \in \{1, \dots, n\}$ generated by the Sampler module, the Evaluator module takes deep features of selected frames $X_e \in \mathbb{R}^{n \times d}$ as input. In order to reduce temporal dimension, we firstly attempted to use 1D convolutional layer to get compact representation $X_{e'} \in \mathbb{R}^n$,

$$X_{e'} = \sigma(\mathcal{F}_{conv1d}(X_e)) \quad (3.8)$$

However, experiments shows that representation produced by 1D convolutional is insufficient to encode all n selected frames, which makes training of the Evaluator module more difficult to converge. Gated recurrent unit was proposed in 2014 [CGCB14] as a simpler variant of LSTM(Long Short Term Memory). By maintaining reset gate and update gate, GRU is able to summarize the temporal information between n selected frames and aggregate latent representation $X_{e'}$. The recursive computations of activation of of GRU are as follows:

$$\begin{aligned} r_t &= \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \\ z_t &= \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \\ n_t &= \tanh(W_{xn}x_t + b_{xn} + r_t \odot (W_{hn}h_{t-1} + b_{hn})) \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot n_t \end{aligned} \quad (3.9)$$

where σ denotes the activation function and \odot denotes element-wise production.

Afterwards, a 3-layer MLP is adopted to get the final effectiveness score $S \in [0, 1]$.

3.5 Training and Implementation Details

3.5.1 Training

Since our proposed module contains multiple modules and no ground truth of noisy frames are available, successfully training the Sampler module and the Evaluator module with only the action recognition loss is not straightforward. Thus we introduce the training strategy we adopted to effectively train the Sampler and Evaluator modules. Our training follows a weakly supervised manner: we only use the action recognition result from the Recognition module as supervision. We show in experiments that even though the ground truth of noisy frames are not given, our sampler module could still localize informative frames and filter out noisy frames.

Initializing the Sampler Module As shown in previous works [LYR15], compared with correct recognition of nouns in actions, it's more difficult for existing action recognition models to accurately recognize the verbs in actions. Take action 'take cup' as an example, existing models can predict the noun 'cup' with high confidence, but they are feeble in distinguishing 'take' from other verbs like 'put' and 'open'.

Our experiments results also verify this observation. We suppose this kind of observation can be attributed the fact that existing models cannot adequately encode temporal information in deep features. However, with the assist of powerful CNN networks, spatial information is better included in deep features. To encourage the Sampler module to learn more temporal information, we propose to use verb recognition task to initialize the Sampler module.

Given ground-truth action verb labels v and predictions \hat{v} produced by the Sampler module, we use standard multi-class classification cross-entropy loss to initialize the Sampler module. We refer to this loss as \mathcal{L}_S :

$$\mathcal{L}_S = - \sum_{i=1}^{|v|} v_i \log(\hat{v}_i) + (1 - v_i) \log(1 - \hat{v}_i) \quad (3.10)$$

Training Recognition Module After training the Sampler module, we take selected frames as the input of the Recognition module. Given ground-truth action labels y , we also minimize the standard cross-entropy loss \mathcal{L}_R to optimize the parameters of the Recognition module:

$$\mathcal{L}_R = - \sum_{i=1}^{|y|} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (3.11)$$

where y and \hat{y} denote truth action labels and predictions respectively.

Training the Evaluator Module In this stage, we fix the Sampler module and the Recognition module, and then train the Evaluator module to generate effectiveness score $S \in [0, 1]$. When selected frames are easier for Recognition module to make confident recognition, the effectiveness score will be closer to 1.

Taken selected frames as input, the Recognition module produces predictions \hat{y} which contains classification possibilities over all action classes. When the possibility of ground-truth action class \hat{y}_{gt} is larger, the Recognition module will be more likely to give right result. Obviously, \hat{y}_{gt} can be contemplated as self-supervision for training the Evaluator module.

For the sake of better frame selection and higher learning ability of the Evaluator module, we set a threshold τ to quantify \hat{y}_{gt} :

$$\hat{y}_{gt} = \begin{cases} 0 & \hat{y}_{gt} < \tau \\ 1 & \hat{y}_{gt} > \tau \end{cases} \quad (3.12)$$

Many people may naturally regard the training of Evaluator module as a 0-1 classification problem after seeing the above equation. However, binary classification is too simple to learn a meaningful score. For example, score with value 0.55 and score with value 0.95 will be treated equally in 0-1 classification problem, but actually they are quite different as for a score.

Therefore, we treat the training of Evaluator module as a regression problem rather than a classification problem. Moreover, we introduce weighted L1 loss with weight w_E to alleviate sample imbalance.

$$\mathcal{L}_E = \sum w_E |S - \hat{y}_{gt}| \quad (3.13)$$

Adapting the Sampler Module In order to guide the Sampler module to select frames that benefit final action recognition, we fix the trained Evaluator module and Recognition module and then use all three loss mentioned above to form loss \mathcal{L}_{adap} :

$$\mathcal{L}_{adap} = \lambda_S \mathcal{L}_S + \lambda_R \mathcal{L}_R + \lambda_E \mathcal{L}_E \quad (3.14)$$

Then we optimize the parameters in Sampler module to minimize the loss \mathcal{L}_{adap} . More detail of loss used to adapt Sampler module can be found in Section 5.1.

3.5.2 Implementation details

We use PyTorch [PGC⁺17] to implement our model. For the Sampler module, all the input frames are converted to gray-scale images and resized to 64×48 , and the feature size d is set as 512. All the input images given to Recognition modules are resized to 320×240 . The Evaluator module consists of one 1-layer GRU with 512 output channels, followed by a 3-layer MLP with output channels to be $\{1024, 256, 1\}$ respectively. For all experiments, N frames correspond to 64 frames. However, n frames can be 4/8/16 frames for different experiments.

About the threshold τ , we set it as 0.65 for ResNet I3D backbone, and set it as 0.15 for ResNet101 backbone and I3D backbone. With regard to w_E in \mathcal{L}_E , we use $w_E=[1,0.85]$ when training the Evaluator module, and change it to $w_E=[1,0.5]$ when adapting the Sampler module. λ_S, λ_R and λ_E are all set to 1 in adapting the Sampler module stage.

We use the Adam[KB14] optimizer to update all the parameters. For initializing the Sampler module and training the Recognition module, the learning rate is set as 1e-3 and will be decayed by 0.1 at epoch [8, 14, 18]. For training the Evaluator module and adapting the Sampler module, the start learning rate is set as 5e-4, and will be decayed by 0.1 at epoch 10.

4 Experiments

In this section, we first describe the dataset we use to evaluate our proposed model. Then we introduce the experimental setup of our experiments in detail. We report the anti-noise results and the action recognition performance in the following two sections. In the end, we show the ablation study result to validate the effectiveness of different modules in our proposed model.

4.1 Dataset

In this work, we use the Extended GTEA Gaze+ dataset [LLR18] (EGTEA Gaze+) to evaluate our models. The dataset is composed of 29 hours of first-person videos collected by head-mounted cameras in a naturalistic kitchen scenario. Figure 4.1 shows representative frames selected from the dataset. Currently, EGTEA Gaze+ dataset is the standard as well as one of the largest egocentric dataset widely used by the research community.

EGTEA Gaze+ dataset consists of 86 unique sessions which come from 32 subjects performing 7 different recipes such as making a pizza or cooking a cheese burger. The video frames are collected at 24 fps with resolution 1280×960 . For video clips which contain actions, action annotations are given in the form of verb+noun pair, such as ‘put + cup’, ‘open + condiment container’. In total, the dataset contains 106 different action classes, 19 verb classes and 51 noun classes.

With regards to split training and testing data for action recognition task on trimmed dataset, we follow the same train-test split of Li *et al.* [LLR18] to split the data without overlapping. This train-test split finally forms 8299 clips for training and 2022 for testing. For the action recognition results, we report mean class accuracy at the clip level as the final action recognition result following [LLR18].

4.2 Experimental Setup

4.2.1 Recognizing interrupted actions

Unlike trimmed videos in normal dataset of third-person videos, first-person videos (like vlogs on YouTube) in real-life setting are much more casual and noisy, which

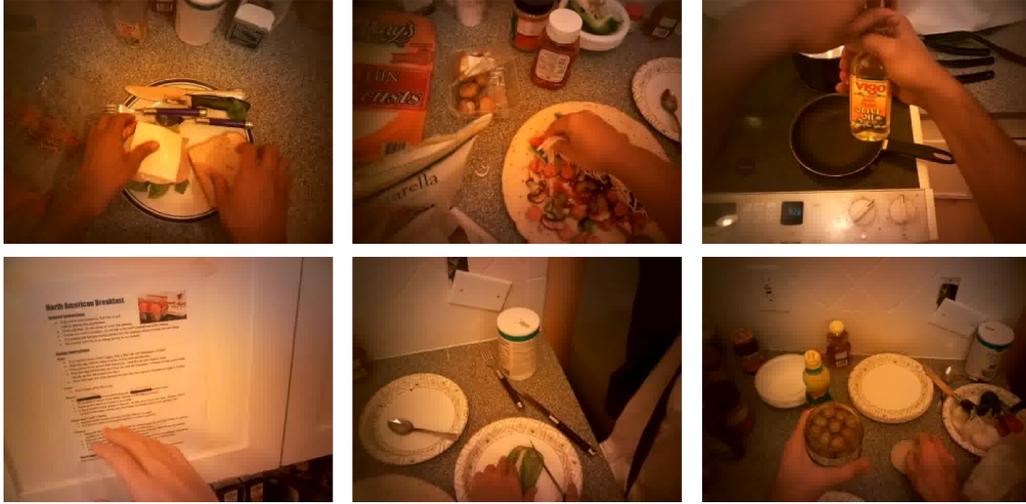


Figure 4.1: Example images in EGTEA Gaze+ dataset. Videos in EGTEA Gaze+ dataset are collected by head-mounted cameras in a naturalistic kitchen scenario.

brings great challenges to correctly recognizing the interrupted actions within. We refer the *action irrelevant frames* in a video clip as noise. In this work we mainly take into consideration the following two conditions of interrupted actions:

Subject-irrelevant actions For example, someone may suddenly look around when she or he is taking the bowl. This is especially common for the camera wearers who is not very familiar with cooking. Since this ‘look around’ action is not relevant with the subject-related action ‘take bowl’, it will bring negative impacts on model performance. If these two actions are mixed and fed as input together the model would have a substantial chance to produce incorrect output.

Outlier frames Due to the lack of professional camera holder, outlier frames may also exist because of the egomotion of the first-person camera. So outlier frames such as background images may appear unexpectedly in continuous actions. And these kinds of outlier frames confuse the model and hinder the training process.

Both of these conditions will have adverse effects on action recognition performance, so we propose our Sampler-Evaluator model to reduce this negative influence. However, since video clips in EGTEA Gaze+ dataset are already manually trimmed, most of subject-irrelevant actions and outlier frames are removed. In order to simulate real-life interrupted actions in egocentric videos, we add noisy frames in trimmed videos and then validate the effectiveness of our model on it.

For the sake of scene consistency, we randomly choose none-action frames from leftover parts (which do not have action annotations) in EGTEA Gaze+ dataset original videos to form noisy frames set \mathcal{A} . Among all 24217 noisy frames, 25%

of them are used for testing and 75% for training, following the train/test ratio in trimmed dataset. Figure 4.2 shows examples of frames that we used in add noise experiments.



Figure 4.2: Example of none-action frames. We randomly choose none-action frames without action annotation from original videos to build noisy frame set \mathcal{A} .

We here describe the strategy of adding noisy clips into trimmed clips: given trimmed video clip with length N , we first randomly divide it to 4 small snippets with arbitrary length. We then randomly choose 2 snippets and replace them with a noisy clip from set \mathcal{A} which is of the same length. Finally, we build mixed input clips with around 50% noisy frames.

4.2.2 Localizing informative frames in super noisy videos

Although the previous experiment on recognizing interrupted actions randomly replace 2 clean snippets with noisy snippets to simulate real-life conditions, it may cause noisy frames to show up intensively in the head or tail of input, which leads to loss of location diversity as well as the dumbing down robustness of model. To further validate the anti-noise ability of our proposed model, we design a more challenging experiment setting: localizing noise frames experiments on super noisy videos.

Given a *noisy* video clip (non-action clip) with length N and a pre-defined noise rate r ($r > 0.5$), we randomly choose $N \times (1 - r)$ frames from clean trimmed clip, and insert them in random locations of noisy frames to build the synthetic super noisy videos. As for the action recognition task on super noisy videos, we report both mean class accuracy and final noise rate in selected frames.

4.3 Recognizing Interrupted Actions

4.3.1 Comparison with baseline methods

In order to validate whether the Sampler module can select informative frames for better recognition, and in the mean time to test the anti-noise ability of our model, we conduct experiments of recognizing interrupted actions following the experiment settings described in the previous section.

There are two previous methods closely related to our work: SCSampler [KTT19] which aimed at sampling salient clips to achieve efficient action recognition; and TAGM (Temporal Attention-Gated Model)[PBTM17] which works on robust action classification by giving higher weight to relevant frames. Of the two methods, the first method relies on ground-truth selection generated by applying brute-force search, which is essentially different from our weakly-supervised method with only ground-truth action classes are used as supervision. Therefore, we only compare our approach against TAGM.

To ensure the number of total informative frames used for action recognition to be the same, our Sampler network takes N frames as input and select 16 of them as the input to the Recognition module. For fair comparison, since the noise rate of input noisy videos are around 0.5, TAGM will take 32 frames as input. Thus, the number of effective frames for action recognition are both 16 for our method and TAGM.

Both TAGM and our model employ ResNet101 as the backbone to extract visual features from the input frames. In addition, we build another baseline: we use ResNet101 trained on clean trimmed videos and uniformly samples 16 frames from clean trimmed videos as input. We introduce this baseline as the **Oracle** of action recognition performance.

Model	Recognition Acc (%)	selected frames noise rate
TAGM [PBTM17]	39.42	/
Proposed	49.16	0.0221
Oracle	52.77	/

Table 4.1: Action recognition performance comparison with baseline methods. We validate whether our proposed Sampler-Evaluator model can select informative frames on recognizing interrupted actions task. Given interrupted actions with around 50 % noisy frames, our Sampler module selects 16 frames from them for Recognition module to generate final recognition results. Recognition module trained on trimmed clean videos and uniformly sampled 16 clean frames as input is viewed as oracle of action recognition performance.

Quantitative results are shown in Table (4.1). Given input noisy videos with noise rate around 0.5, our proposed modules can distinguish informative frames from noisy frames and remarkably reduce the noise rate in selected frames. With the same number of informative frames used for recognition, our Sampler-Evaluator modules can greatly improve the action recognition accuracy compared with TAGM baseline.

Based on the above results, we can conclude two points. Firstly, although the TAGM method can achieve robust action recognition to some extent by adding temporal attention, it's turns out that their model can't recognize interrupted actions with around 50% noisy frames. We think this is because attention used in TAGM may help model to put emphasis on salient frames, but it can't completely eliminate the effect of noise frames. As a result, when the noisy frames becomes dominant, their method fails to function well. Differently from their method, the output of our Sampler module are discrete selected frames, so the frames that are not selected will have no impact on the final recognition. Secondly, taken the advantage of Sampler-Evaluator modules, our whole model can obtain similar action recognition performance as in clean trimmed videos.

4.3.2 Cooperating with multiple recognition backbones

Since our Sampler-Evaluator modules work in a plug-and-play fashion, in this subsection we verify the cooperation between our Sampler-Evaluator with different Recognition modules as backbone. Here we choose three most commonly used action recognition models as the backbone of our Recognition module:

ResNet101 A common problem in CNNs is that as the network becomes deeper, it becomes easier to overfit, which greatly reduces the learning ability of the model. ResNet utilizes residual blocks to alleviate overfitting problem. With the assist of residual block, ResNet can grows deeper and thus generate more representative features.

I3D [CZ17] I3D is the short name of Inception 3D network, which extends 2D spatial convolutions to 3D spatial-temporal convolutions. Compared with 2D convolutional networks, I3D can better aggregate and utilize the temporal information.

ResNet I3D [WGGH18] ResNet I3D is Inception 3D convolutional network with ResNet as it backbone. As it combine bottleneck structure the with 3d convolution, it also inherited the advantages of both methods and can generate better spatial-temporal deep features for video understanding. Experiments show that ResNet I3D works as a strong baseline for both third-person action recognition tasks and first-person action recognition tasks. However, with the complexity of structures, the calculation consumption also increases dramatically. As a compromise, we use a relatively light-weight ResNet50 network as the backbone.

For each backbone, we define backbone trained on trimmed videos as *clean backbone*, and accordingly we call backbone trained on videos with interrupted actions as *noisy backbone*. Under the premise that the number of frames given to Recognition module is the same, we consider following alternatives:

- clean backbone on clean input (**Oracle**)
- clean backbone on noisy input
- noisy backbone on noisy input
- clean backbone + Sampler-Evaluator on noisy input (**Proposed**)

Obviously, the first alternative works as the Oracle of action recognition performance.

Alternatives	Input	Action recognition Acc (%)		
		ResNet I3D	I3D	ResNet 101
clean backbone	noisy videos	41.89	34.27	39.17
noisy backbone	noisy videos	52.13	44.26	41.89
clean backbone+ours	noisy videos	57.74	48.22	49.16
clean backbone (Oracle)	trimmed videos	60.68	52.18	52.77

Table 4.2: Quantitative comparison on cooperating with different backbones. We validate the ability of our proposed Sampler-Evaluator modules on recognizing interrupted actions task. We compare our method with 3 alternatives, and the first alternative which given trimmed videos as input works as the oracle of action recognition performance. For all the alternatives, the number of input frames given to backbones are set as 16.

Quantitative results are shown in Table 4.2. The clean backbone with our Sampler-Evaluator module clearly outperforms clean backbone without our modules and noisy backbone trained on noisy videos. Also, for all backbones our proposed model performs consistently better than the other two alternatives, indicating that our plug-and play Sampler-Evaluator modules cooperate well with all the backbones. By comparing the performance of different backbones, it can be seen that ResNet101 network itself is more robust on recognizing interrupted actions. And due to the disruption of information coherence in time dimension, two 3D convolution based network are badly affected. Fortunately, our Sampler-Evaluator modules can reduce the negative impact of action-irrelevant frames and restore them to a level similar to that in clean trimmed videos.

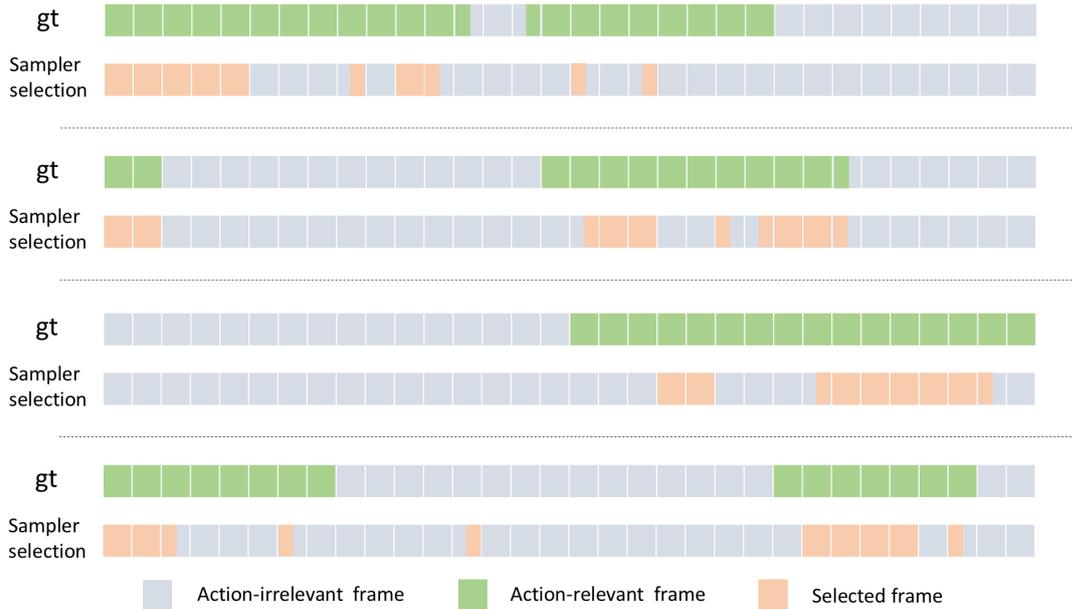


Figure 4.3: Visualization of frame selection results for recognizing interrupted actions. For the sake of simplicity, each unit of square represents two consecutive input frames. Green and gray denote action-relevant frames and action-irrelevant frames respectively. And orange represents frames selected by our Sampler module.

4.3.3 Visualizing frame selection

In this subsection, we visualize the frame selection generated by our Sampler module. Figure 4.3 shows examples of frame selections on noisy input videos used for recognizing interrupted actions task. For the sake of simplicity, one unit of square in the figure represents 2 consecutive input frames. We can clearly observe that our Sampler module is fully capable of distinguishing action-relevant frames from action-irrelevant frames. More importantly, the distribution of frame selection is also diverse, which means frame selection is not simply based on relative position in all input frames. In addition, from the bottom example in Figure 4.3, we can see that the Sampler module may also incorrectly choose noisy frames as one of selected frames, which demonstrates that our model has not yet achieved 100% accuracy.

4.4 Localizing Informative Frames in Super Noisy Videos

To further validate the ability of finding informative frames of our Sampler-Evaluator model, we randomly insert very few useful frames in noisy frames to build super

noisy videos to evaluate our proposed model. Similarly, we solve this problem in a weakly-supervised setting, where only ground-truth action classes of Recognition

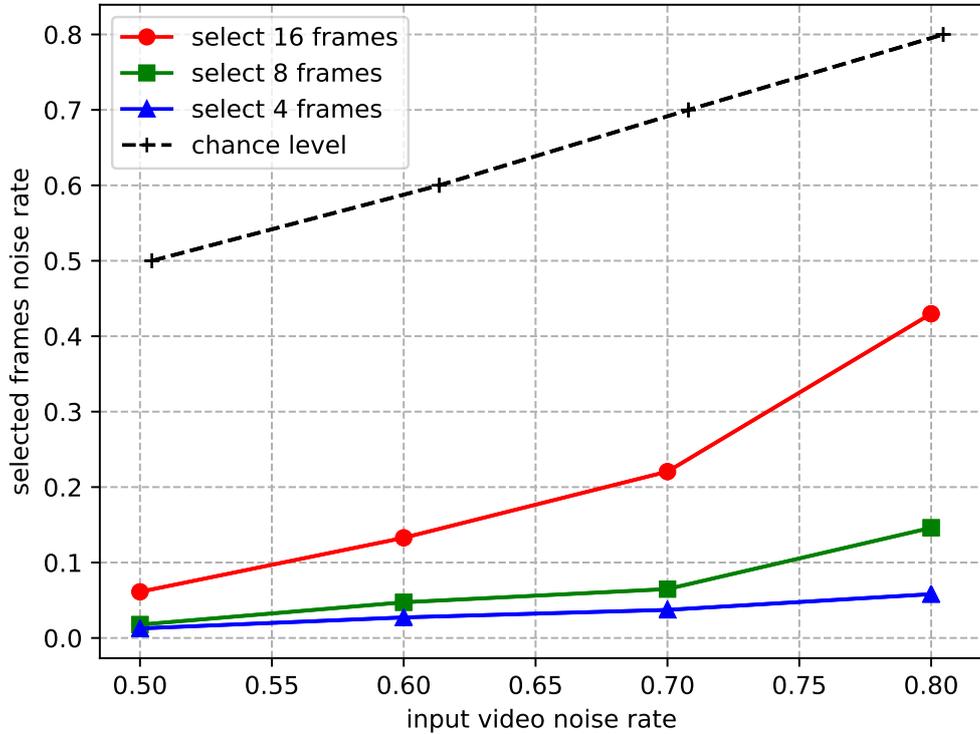


Figure 4.4: Quantitative results of selected frames noise rate. The abscissa corresponds to total input video noise rate, and the ordinate represents the noise rate of selected frames generated Sampler module. Blue, green and red lines show the results of selecting 4/8/16 frames from all input frames. And the black dotted line demonstrates noise rate from random sampling.

For N input frames with data noise rate ranging from 0.5 to 0.8, we report the selected frames noise rate of our model. Without loss of generality, we set the our Sampler module to select 4/8/16 frames from N input frames.

Figure 4.4 depicts qualitative results of selected frames noise rate. We confirm that the noise rate in our selections is significantly lower than data noise rate, which means our Sampler-Evaluator module is capable of localizing and finding the informative frames even in super noisy videos. In the case of selecting 4 frames from N input noisy frames, we can achieve selection noise rate results well below 0.1 even with data noise rates as high as 0.8. For selecting 8 frames from input frames with noise rate, only about one frame will be wrongly selected as a noisy frame. This strongly indicates that our proposed model is very effective in finding noisy frames and selecting informative frames for better action recognition.

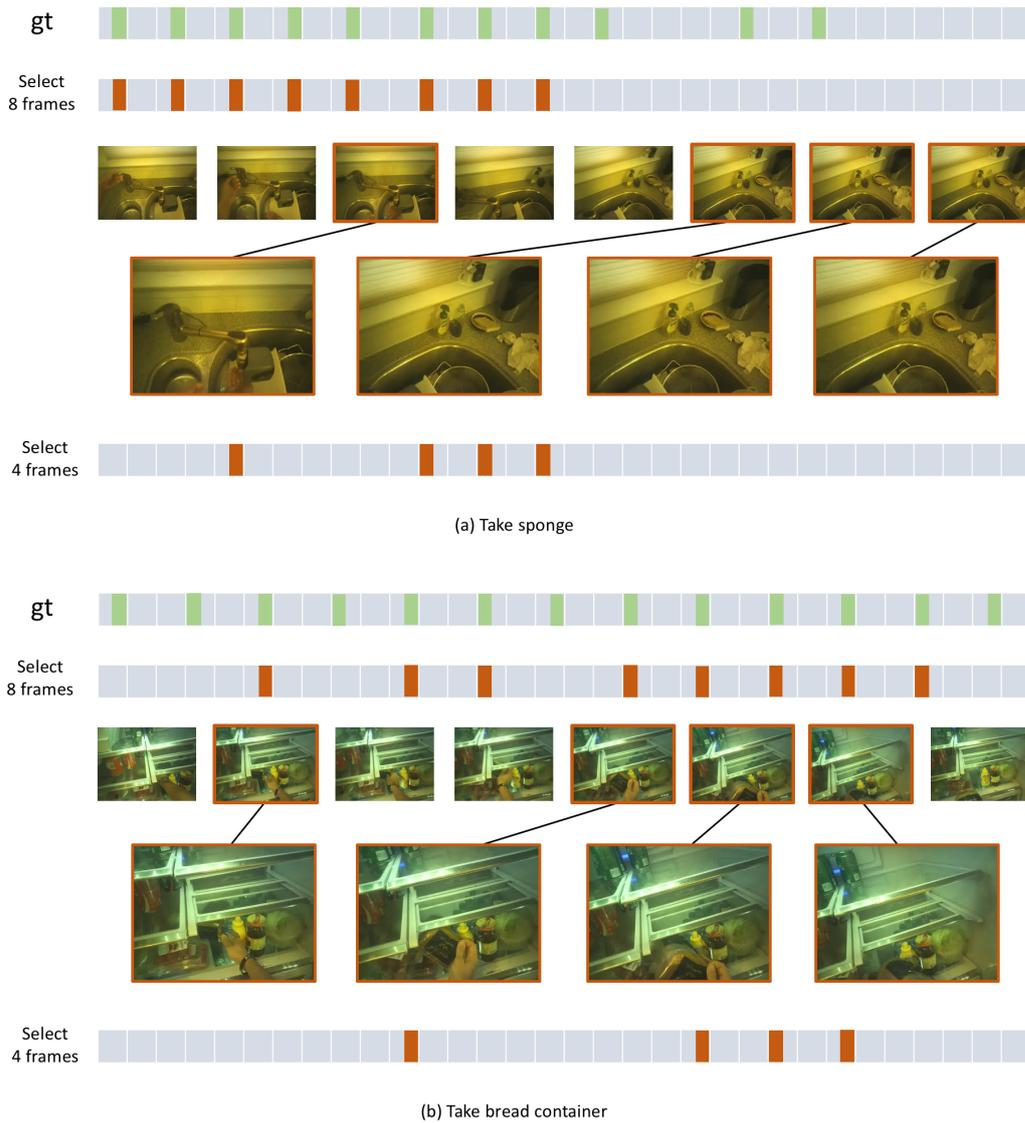


Figure 4.5: Visualization of frames selection results on super noisy videos.

Each unit of square represents two consecutive input frames. Green, gray and red denote action-relevant frames, action-irrelevant frames and frames selected by our Sampler module respectively. Frames connected with black lines are the same. In (a), the action is ‘Take sponge’. Our modules give priority to frames which contains the key object sponge. In (b), for recognizing action ‘Take bread container’, the Sampler module sacrifices a few details of taking process and select more representative frames which could obviously find bread in it.

In the mean time, we also find that the when selecting 16 frames, localization performance decreases in the case that data noise rate is greater than 70%. We suspect this phenomenon can be attributed to the fact that when data noise rate is greater than 0.7, the total number of informative frames in all N input frames is less than 16. As a consequence of this, our Sampler module has to choose some noisy frames as selection.

For better analyze the performance of our Sampler-Evaluator model, we also qualitatively visualize examples of frame selections results on super noisy videos. Both the selection results of selecting 4/8 frames and the corresponding frames can be found in Figure 4.5.

From Figure 4.5 we can see that our Sampler-Evaluator modules can accurately locate informative frames even if informative frames account for only 20% of the whole input. Additionally, by comparing 4 selected frames and 8 selected frames, we can draw a conclusion that our modules have the ability to select salient frames for better action recognition. As shown in Figure 4.5(a), in order to enable Recognition module to correctly recognize action ‘Take sponge’ as far as possible when only four selected frames are input, our modules finally give priority to frames which contains key object sponge.

Similar conclusion can be obtained from Figure 4.5(b). For recognizing action ‘Take bread container’ with limited frames, the Sampler module sacrifices a few details of taking process and select more representative frames which could obviously find bread in it.

4.5 Ablation Study

To validate the effectiveness of each module of our proposed model, we conduct ablation study with the following baselines:

- **Recognition module only (RM)**: without loss of generality, we use recognition module with ResNet I3D as backbone trained on clean trimmed videos as one of the baseline. In the test stage, the input frames are uniformly sampled from noisy videos. The performance of this baseline reveals the anti-noise ability of ResNet I3D network itself.
- **Recognition module + Sampler module (RM+SA)**: In this baseline, the Sampler module is initialized with verb recognition task, which only uses ground-truth verb class as supervision. As in previous baseline, Recognition module is also trained on clean trimmed videos, but it takes selected frames from the Sampler module as input. We build this baseline to validate the effectiveness of our Sampler module.
- **Recognition module + Sampler module + Evaluator module (proposed, RM+SA+EV)**: First of all, we fix Recognition module and initialized Sampler module to train Evaluator. Then Sampler module is adapted

according to the guide of Evaluator. By adding Evaluator module, we enable end-to-end training of the whole model. The performance of this baseline shows whether Evaluator module is effective to lead Sampler module towards selecting more informative frames for Recognition module.

For simplicity of writing, we call each baseline as RM, RM+SA and RM+SA+EV respectively.

4.5.1 Results on recognizing interrupted actions

Module	Selection noise rate	Recognition Acc(%)
RM	0.5016	41.89
RM+SA	0.0389	57.34
RM+SA+EV	0.0221	57.74

Table 4.3: Ablation study for different parts of our model on recognizing interrupted actions. We conduct recognizing interrupted actions task on noisy videos with around 50% noisy frames. For different combination of our modules, both selection noise rate and final action recognition accuracy are reported.

Quantitative results of the ablation study on recognizing interrupted actions are shown in Table 4.3. We can observe our Sampler module can distinguish action-relevant frames from action-irrelevant frames and select informative frames for Recognition module. Taken selected frames generated by Sampler module, Recognition module gains a substantial increase in final recognition accuracy. At the same time, Evaluator module can slightly improve the frame selection performance of Sampler on account of selection noise rate and final recognition accuracy. However, since the selection noise rate of Sampler module itself is already very low, the improvement brought by Evaluator module is not obvious.

4.5.2 Results on localizing useful frames from super noisy videos

Quantitative results of the ablation study on localizing informative frames in super noisy videos are shown in Table 4.4. In this table the noise rate is set as 0.8 and the Recognition module (RM) is ResNet I3D. From the results we can conclude that by using ResNet I3D network alone (the first row), the network has barely anti-noise ability. As a result, the action recognition accuracy becomes very low - only around 14% given super noisy videos. Also, the comparison of the result with/without the Sampler module demonstrates the effectiveness of our Sampler module trained on action verb recognition task.

Module	Selection noise rate			Action recognition Acc (%)		
	4 frames	8 frames	16 frames	4 frames	8 frames	16 frames
RM	0.8047	0.8046	0.8045	14.19	13.24	13.88
RM+SA	0.1327	0.2246	0.4603	44.59	43.01	34.08
RM+SA+EV	0.0581	0.1454	0.4297	47.50	49.06	41.72

Table 4.4: Ablation study for different parts of our model on localizing useful frames in super noisy videos. We validate the effectiveness of Sampler module and Evaluator module on noisy videos with 80% noisy frames. For selecting 4/8/16 frames from noisy videos, both noise rate of the selection and the action recognition accuracy with selected frames as input are reported.

In addition, our experiment results prove that after adapting Sampler module through end-to-end training it can select more informative frames and thus reduce noise rate as well as improve recognition accuracy, which validate our Evaluator module is capable of enabling gradient back propagation and guiding Sampler module.

5 Discussion

5.1 Impact of Different Kinds of Loss in Sampler Adaption

During our experiments, after training the Evaluator module, we try different kinds of loss to adapt the Sampler module:

- **Replace ground-truth class recognition possibility with 1 in \mathcal{L}_E**

Since we introduce the Evaluator module to guide the Sampler module to select frames that benefit final action recognition, the value of ground-truth class recognition possibility is the most important reference for adapting Sampler module. So we propose to replace that values with 1 to form new Evaluator loss \mathcal{L}_{adap} and adapt Sampler module to minimize this loss.

$$\mathcal{L}_{adap} = \sum w_E |S - 1| \quad (5.1)$$

However, although replacing ground-truth possibility with 1 sounds theoretically feasible, it turns out that in actual experiments the Sampler module fits in one epoch but doesn't optimize in the right direction. That means by using this loss Sampler module hardly learn any useful things.

- **Combination of Evaluator loss and Recognition loss**

We also attempt to use the combination of Evaluator loss and Recognition loss to adapt the Sampler module. Theoretically, Recognition loss \mathcal{L}_R will encourage higher possibility of ground-truth action class, which will thus affect Evaluator loss \mathcal{L}_E and then indirectly guide the selection process of the Sampler module.

$$\mathcal{L}_{adap} = \lambda_R \mathcal{L}_R + \lambda_E \mathcal{L}_E \quad (5.2)$$

Unfortunately, experiments shows that combination of these two losses is too implicit to optimize the Sampler module.

- **Combination of Evaluator loss, Recognition loss and Sampler loss**

After the above attempt failed, we further try to add Sampler loss \mathcal{L}_S to Eq.5.2.

$$\mathcal{L}_{adap} = \lambda_S \mathcal{L}_S + \lambda_R \mathcal{L}_R + \lambda_E \mathcal{L}_E \quad (5.3)$$

Experimental results of clean Recognition module trained on trimmed videos with initialized Sampler module demonstrates that Sampler loss itself is able to guide the Sampler module to select meaningful frames. Therefore, the addition of Sampler loss can assist in optimizing Sampler module in right direction. Experimental results also prove our assumption.

Based on above attempts, we finally decide to use the combination of Evaluator loss, Recognition loss and Sampler loss. We fix the Recognition module and the Evaluator module to minimize \mathcal{L}_{adap} by adapting the Sampler module.

5.2 Comparison between Different Attention Mechanisms

In Section 3.3, we explain details of two attention mechanisms: self-attention and TAGM. To have a better knowledge about two attention mechanisms, we discuss the difference between selection noise rate and final action recognition accuracy when adopting these two attention mechanisms to calculate attention scores for N input frames.

Attention mechanism	Selection noise rate	Action recognition Acc (%)		
		ResNet I3D	I3D	ResNet 101
Self-attention [VSP ⁺ 17]	0.0381	57.34	47.02	49.05
TAGM [PBTM17]	0.0257	56.25	46.83	48.81

Table 5.1: Quantitative comparison on using different attention mechanisms. We compare the selection noise rate of initialized Sampler with two different attention mechanisms on recognizing interrupted actions task. Final action recognition results of three backbones with different initialized Sampler are also reported.

On recognizing interrupted experiments, we report the selection noise rate of the Sampler module with two different attention mechanisms initialized by the same verb recognition task as well as the action recognition of the Recognition module with initialized Sampler module. For the sake of fairness, all the parameters and backbones are the same except for applying different attention mechanisms.

According to Table (5.1), we observe that the noise rate of selected frames given by TAGM attention mechanism is lower than self-attention mechanism. However, for all three backbones, Sampler module with self-attention mechanism consistently outperforms the other one with TAGM attention mechanism in action recognition accuracy. We suppose the ability to determine ‘keep’ and ‘neglect’ of TAGM accounts for lower selection noise rate. However, this ‘keep’ and ‘neglect’ ability also

lead to more emphasis on absolute relations between frames and aggregated clip features rather than paying attention to relative relations between N frames. On the contrary, self-attention mechanism has more advantages in getting the relative relations between frames, so it can select relatively more informative frames from N frames and thus obtain higher recognition accuracy. In the rest of thesis, all the reported results are based on self-attention mechanism.

5.3 Limitation of Our Model

While the proposed approach presents strength in selecting informative frames from noisy input videos, there are some limitations of our model. First, each module is very dependent on other modules and has to be trained one by one. The training method of our model is explained thoroughly in Section 3.5. Currently we need to follow previously described order to train the whole model.

In addition, the training of the Evaluator module may face data imbalance problem. For example, if the trained Recognition module can achieve 80% recognition accuracy on training set, then around 80% of the Evaluator ground-truth will be 1, which is probably 4 times of ground-truth 0. Although we have introduced weighted L1 loss and threshold τ to alleviate the impact of ground-truth imbalance, but in extreme cases the training of the Evaluator module still get affected by this problem. One possible solution is early stopping in the training process of Recognition module, leaving the Recognition module not so overfitting. We will keep finding other better solutions in the future.

6 Conclusion and Future work

In this thesis, we propose a new method for selecting informative frames from noisy video clips to better perform action recognition. Action recognition is one of the primitive research field in computer vision. It attracts significant research attention due to the wide application ranging from surveillance to human behaviour analysis. Although recent deep learning techniques have made great success in action recognition, existing methods still cannot work well on real-world videos where many noisy frames exist.

Built on the success of existing methods for action recognition, we propose a Sampler-Evaluator model for filtering out noisy frames from the input. Our proposed model works in a plug-and-play fashion, so it could be applied on top of any existing models for action recognition. To be specific, our model can take as input unprocessed noisy video clips and output only a few informative frames. As most state of the art models for action recognition only needs a few informative frames however decreases significantly when the input frames are not carefully selected, our model provides a complementary functionality against the existing models that could select frames in order to better perform action recognition in more general and natural videos.

Other than the recognition module which essentially could be any models for action recognition, the model we proposed contains two major modules as components: the Sampler module and the Evaluator module. The sampler network takes as input long and noisy unprocessed video clips, and outputs a sequence of selected informative frames. Since the ground-truth of noisy frames are unknown, we use an evaluator network for better training the sampler. The evaluator network evaluates the quality of the selected sequence, using the information that how well are the selected sequence recognized by existing action recognition models. The role of evaluator acts like a teacher that provides feedback for better training the sampler network. During the inference step, the evaluator is discarded.

We show in experiments that by adding the sampler we can better recognize action from noisy video inputs, and that by adding the evaluator in training it is able to boost the final action recognition performance even more. We also analyzed the limitations of our method and will tackle those in our future work. The biggest limitation of our method is the difficulty in training the whole network. While with our proposed training strategy, we can successfully train the sampler and evaluator and in the mean time optimizing the performance of the action recognition, the training procedure is complicated with many hyper-parameters to be carefully tuned. In the future we will be continue working on this to make the training more straightfor-

ward. We will also collect a novel dataset simulating the interrupted actions, for better evaluation and designing for new models.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Yoichi Sato for the continuous support of my Master's study and research. He offered us flexible environment to explore our areas of interest, and when we are stagnant he will always guide us in time. His dedication to refinement and the scrupulous attention to research influenced us imperceptibly, and he will always be our model in research. I'm impressed by his patience, motivation, enthusiasm, and immense knowledge, and I could not imagine a better advisor and mentor during my study.

Besides my advisor, I would like to thank associate Prof. Yusuke Sugano. Although we didn't get along for a long time, he provided me with sincere and useful suggestions at every meeting. In addition to academic advice, he also taught me a lot about how to be a qualified researcher and how to be a modest and polite person.

Special thanks should also be expressed to co-author Yifei Huang. Discussion with he is always pleasant and inspiring. Also, his patience, tolerance and positiveness have taught me a lot. I'm really grateful that he has been supporting me spiritually throughout my study.

My sincere thanks also go to all of my labmates in Sato Lab, for the stimulating discussions, for the kindness of everyone and for the fun we have had in the last two years. I will remember deeply in my heart how friendly they are and wish them all the best in the future.

Last but not the least, I would like to express my heartfelt thanks to my parents. Without their steadfast support and continuous consideration, I would not be so optimistic about my life and study. Thank you for always standing with me.

References

- [BEG⁺17] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *The British Machine Vision Conference (BMVC)*, volume 2, page 7, 2017.
- [CGCB14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [CHCNG16] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [CKL⁺18] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Multi-fiber networks for video recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [CLS15] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [CVS⁺18] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [CWPQ14] Zhuowei Cai, Limin Wang, Xiaojiang Peng, and Yu Qiao. Multi-view super vector for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [CZ17] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [DBPV05] Thi V Duong, Hung Hai Bui, Dinh Q Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005.

- [DDS⁺09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [DFS⁺18] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [FGMR09] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2009.
- [FGO⁺15] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [FPW16] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [FPZ16] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [GCGS14] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.
- [GGCN19] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. Mac: Mining activity concepts for language-based temporal localization. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019.
- [GGRVG14] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.
- [GGVG15] Michael Gygli, Helmut Grabner, and Luc Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [GHS13] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013.

- [GSK⁺16] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2016.
- [GYZ⁺16] Chuang Gan, Yi Yang, Linchao Zhu, Deli Zhao, and Yueting Zhuang. Recognizing an action using its name: A knowledge-based approach. *International Journal of Computer Vision (IJCV)*, 2016.
- [HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [JVGJ⁺14] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek. Action localization with tubelets from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [JXYY12] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.
- [KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [KL14] Vadim Kantorov and Ivan Laptev. Efficient feature extraction, encoding and classification for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [KpVD⁺19] Georgios Kapidis, Ronald Poppe, Elsbeth van Dam, Lucas Noldus, and Remco Veltkamp. Multitask learning to improve egocentric action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2019.
- [KTS⁺14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [KTT19] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. *arXiv preprint arXiv:1904.04289*, 2019.
- [KW16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [KWFS17] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

- [LLL⁺19a] Xin Li, Tianwei Lin, Xiao Liu, Chuang Gan, Wangmeng Zuo, Chao Li, Xiang Long, Dongliang He, Fu Li, and Shilei Wen. Deep concept-wise temporal convolutional networks for action localization. *arXiv preprint arXiv:1908.09442*, 2019.
- [LLL19b] Minlong Lu, Danping Liao, and Ze-Nian Li. Learning spatiotemporal attention for egocentric action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2019.
- [LLR18] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [LYR15] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [MJM⁺18] Michele Merler, Dhiraj Joshi, Khoi-Nguyen C Mac, Quoc-Bao Nguyen, Stephen Hammer, John Kent, Jinjun Xiong, Minh N Do, John R Smith, and Rogério Schmidt Feris. The excitement of sports: Automatic highlights using audio/visual cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.
- [MLT17] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [MMJ⁺18] Michele Merler, Khoi-Nguyen C Mac, Dhiraj Joshi, Quoc-Bao Nguyen, Stephen Hammer, John Kent, Jinjun Xiong, Minh N Do, John R Smith, and Rogerio Schmidt Feris. Automatic curation of sports highlights using multimodal excitement features. *IEEE Transactions on Multimedia (TMM)*, 2018.
- [NCFF10] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.
- [NMY15] Bingbing Ni, Pierre Moulin, Xiaokang Yang, and Shuicheng Yan. Motion part regularization: Improving action recognition via trajectory selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [PBTM17] Wenjie Pei, Tadas Baltrusaitis, David MJ Tax, and Louis-Philippe Morency. Temporal attention-gated model for robust sequence classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

- [PFR17] AJ Piergiovanni, Chenyou Fan, and Michael S Ryoo. Learning latent subevents in activity videos using temporal attention filters. In *AAAI Conference on Artificial Intelligence*, 2017.
- [PGC⁺17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [PWWQ16] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding (CVIU)*, 2016.
- [QYM17] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [RYW18] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [SEL19] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [SJYS15] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [SKS⁺18] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [SL18] Swathikiran Sudhakaran and Oswald Lanz. Attention is all we need: nailing down object-centric attention for egocentric activity recognition. *arXiv preprint arXiv:1807.11794*, 2018.
- [SMP⁺17] Deepika Singh, Erinc Merdivan, Ismini Psychoula, Johannes Kropf, Sten Hanke, Matthieu Geist, and Andreas Holzinger. Human activity recognition using recurrent neural networks. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2017.
- [SVI⁺16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

- [SVSJ15] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [SWC16] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [SZ14] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.
- [TBF⁺15] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [TIL04] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *International Conference on Pervasive Computing (ICPC)*. Springer, 2004.
- [UAM⁺17] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Action recognition in video sequences using deep bi-directional lstm with cnn features. *IEEE Access*, 2017.
- [VKNEK08] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the International Conference on Ubiquitous Computing*. ACM, 2008.
- [VLS17] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.
- [VSP⁺17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [WG18] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [WGGH18] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [WJQ⁺17] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention

- network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [WKSCL11] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [WLLVG18] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [WMD18] Yaming Wang, Vlad I Morariu, and Larry S Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [WQT] Limin Wang, Yu Qiao, and Xiaoou Tang. Mofap: A multi-level representation for action recognition. *International Journal of Computer year=2016, publisher=Springer*.
- [WQT13] Limin Wang, Yu Qiao, and Xiaoou Tang. Latent hierarchical model of temporal structure for complex activity classification. *IEEE Transactions on Image Processing (TIP)*, 2013.
- [WQT14] Limin Wang, Yu Qiao, and Xiaoou Tang. Video action detection with relational dynamic-poselets. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2014.
- [WQT15] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [WS13] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3551–3558, 2013.
- [WXW⁺16] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.
- [XE19] Chengguang Xu and Ehsan Elhamifar. Deep supervised summarization: Algorithm and application to learning instructions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [XSH⁺18] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy

- trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [YMR16] Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [YXL18] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2018.
- [ZAOT18] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [ZCSG16] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.
- [ZGY⁺18] Ming Zeng, Haoxiang Gao, Tong Yu, Ole J Mengshoel, Helge Langseth, Ian Lane, and Xiaobing Liu. Understanding and improving recurrent networks for human activity recognition by continuous attention. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. ACM, 2018.
- [ZHT⁺19] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [ZSB18] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [ZSZZ18] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Mict: Mixed 3d/2d convolutional tube for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [ZXL⁺18] Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. Adding attentiveness to the neurons in recurrent neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

List of Publications

1. Lijin Yang, Yifei Huang, and Yoichi Sato, “Egocentric Action Recognition using Graph Convolution Networks,” In Extended Abstract of Meeting on Image Recognition and Understanding (MIRU), Aug. 2019
2. Lijin Yang, Yifei Huang, Yusuke Sugano and Yoichi Sato, “Egocentric Action Recognition on Noisy Videos, ” In Proceedings of the Pattern Recognition and Machine Understanding (PRMU), Mar. 2020

