Video Action Recognition with Adaptive Zooming Using Motion Residuals

Mostafa Shahabinejad [*1,4], Irina Kezele [†1,3], Seyed Shahabeddin Nabavi [1,3], Wentao Liu [1,3], Seel Patel [1,4], Yuanhao Yu [1,3], Yang Wang [2,4], and Jin Tang[1,3]

[1]Noah's Ark Laboratories, Huawei Technologies, Markham, Ontario, Canada
[2]Concordia University, Montreal, Quebec, Canada

[3]{irina.kezele ,shahab.nabavi,yuanhao.yu,tangjin}@huawei.com
[4]{m.shahabinezhad@gmail.com, liu.wen.tao90@gmail.com,
seel.patel@utoronto.ca,yang.wang@concordia.ca}

Abstract

Motivated by the mechanisms of selective visual attention in humans, we put forward an efficient method for learning spatial attention with adaptive zooming for video action recognition. The learnt module can be used as a plug-in with any 3D CNN action recognition model with clip-level processing. We propose to use relevant motion clues from video frames to adaptively learn input-clip optimal transformations, as these clues are hypothesized to be directly related to the action recognition task. We employ differentiable transformations and samplers and ensure end-to-end system differentiability. We render the proposed module light-weight and computationally efficient, by exploiting the motion information inherently present in compressed videos and readily available at both training and inference time. Highly informative motion-related content of compressed video domain modalities helps further boost action recognition accuracy. Our experimental work demonstrates clear benefits of the proposed method for adaptive spatial zooming and of utilizing the compressed domain for that purpose.

1. Introduction

Substantial evidence from neurobiology and cognitive sciences [2, 7, 26] supports the following conceptualizations of human visual attention: Inputs compete for representation in multiple visually responsive brain systems and the selection takes place through integration across recurrently connected systems. The selected object properties and spatial locations tend to become dominant throughout.

Recent developments in deep neural networks have led to progressive integration of attention mechanisms into multiple disciplines within computer vision, including: image classification [24], image recognition [1], image captioning [42], and video action recognition [28, 37, 9, 3, 20].

We approach this problem from the perspective of video Action Recognition (AR). We observe that it is both scene context/semantics and motion that are representative of the underlying action category [18]. For example, actions like brushing teeth or playing an instrument rely more on the context, while actions like jumping, walking or running are better explained by motion. In this regard, the key assumptions we make are that the Classification-Regions-of-Interest (CROIs) for human AR in videos are, in general, tightly linked to regions of motion-related spatial saliency [19, 18], and that the CROIs are also implicitly related to regions of semantic saliency, since the AR-relevant motion is typically associated with subjects and objects of interest [22]. Consequently, we propose to use motion clues from video frames to guide learning of the pertinent visual regions, more specifically in a form of zooming-in to pertinent input signal CROIs for AR. By scaling the selected CROIs to the original input size, we further achieve a higher apparent CROI resolution.

We realize efficiency, by exploiting the motion information inherently present in compressed videos and readily available at inference time. Capitalizing on sparseness and high informative content of motion vectors and residuals from the compressed video domain, relevant to human AR, we define compact and fast modules to learn the optimal input transformations.

We experimentally prove that our approach results in

*Corresponding author. Work done while an employee at Huawei Technologies Canada.
†Corresponding author at: Noah's Ark Laboratories, Huawei Technologies Canada.

straightforward accuracy improvement for the video AR task, demonstrated on the class of 3D CNN AR methods, with video-clip modeling.

The main contributions of our work are as following:

- A general method for spatial attention by adaptation of video-clip spatial transformations, guided by motion clues, within the context of AR with 3D CNNs and clip-level modeling.
- A light-weight model to learn the optimal transformations for preprocessing input video-clips, utilizing compressed video domain modalities. This model can be used as a pluggable module with any 3D CNN AR model.
- Instructive experimentation, with a particular choice of datasets for method evaluation, to demonstrate the method effectiveness on both datasets with more prominent static and datasets with more prominent temporal characteristics of action categories.
- A variety of ablation experiments, including the validation of the benefits of using compressed video modalities over RGB, for learning optimal input-clip transformations.

2. Related Work

Action Recognition in Videos.

With the emergence of more sophisticated deep learning techniques, several mainstream directions for video action recognition have been proposed, including: a) per-frame modelling with 2D CNNs of the entire video, or of video clips, with: either RGB modality only [10, 31], or RGB modality complemented with a form of an explicit temporal component modeling, where the latter can be accomplished by using optical flow [30, 36], or recurrent neural networks [6, 21]; and b) clip processing with 3D CNNs, to implicitly encode temporal information on the clip level [34, 14, 35]. We focus on 3D CNN category and clip-level video modelling, although our proposed method can be extended to 2D methods and per-frame modelling likewise. Within this framework, we design our adaptive input transformation module as a plug-in with any 3D CNN architecture that models video clips.

Compressed Video Action Recognition. Videos are usually compressed for efficient transmission and storage. To do so, video codecs split the whole video into Groups of Pictures (GoPs) as the basic encoding units, each of which consists of a leading intra-frame (I-frame) followed by a sequence of predicted (P-) and bidirectional (B-) frames. In contrast to the leading I-frame, which is encoded as an independent image, the P- and B-frames in the GoP are represented by their "differences" with respect to some previous reference frames. Such "differences" contain two data types, the motion vectors and the residuals, which provide compact representations for the relative motion between frames in a compressed video. As a result, recent years have seen an increasing interest in leveraging the compressed domain information for efficient action recognition [4, 11, 12, 39, 29]. In particular, many studies show that the freely-available motion vectors and residuals can replace the optical flow as the input to a temporal network to greatly improve the efficiency of two-stream action recognition methods. Our proposed method also takes advantage of the compressed domain information, but is substantially different from the previous works in that: 1) we aim at improving the accuracy rather than the efficiency of existing action recognition models, 2) to help achieve this goal with full potential, we extend the compressed domain setup and propose to work with all the available information, from both the raw and compressed video domain. This way, we jointly make use of the decoded RGB images and the related compressed domain modalities at every video frame. We elaborate on this setup in Section 3.1 below.

Adaptive Action Recognition. A number of adaptive AR approaches have also been proposed lately. Those primarily target computational efficiency, with accuracy improvement as a putative secondary outcome. Some works, for example look into previewing videos with computationally light models, followed by decision making to: either select subsets of informative frames/clips for processing [16, 43, 25], as subsets of the entire video, or to process adaptively by defining learnable policies. The latter produces sequential per-frame decisions, where the decision made is on the required resolution and model capacity depending on the frame content, considering also the context of the so far seen information [41, 23]. AdaFocus [38] achieves computational efficiency by processing smaller input patches of a fixed size, where patch centers are selected with reinforcement learning, over a predefined set over the image grid. Informative image patches are traced through the 2D frame sequence, for patch-based 2D AR modeling. The latter is closest to our approach in that optimal patches are adaptively learnt given the input, but with the following important differences: 1) Our patch size is adapted to the input, contrary to AdaFocus' fixed size patches, 2) We regress continuous patch sizes, 3) We learn the adaptive module simultaneously with the 3D CNN backbone, end-to-end, thanks to differentiable transformations and samplers [13], 4) We are primarily motivated by accuracy improvement, while maintaining the same overall model efficiency: we resize our adaptive CROIs and model them at a higher resolution input size, effectively increasing the apparent input model resolution (and to note is that our light-weight model that uses compressed video modalities adds only negligible computational load), 5) We learn to adapt to short clip-inputs although our method can be extended to

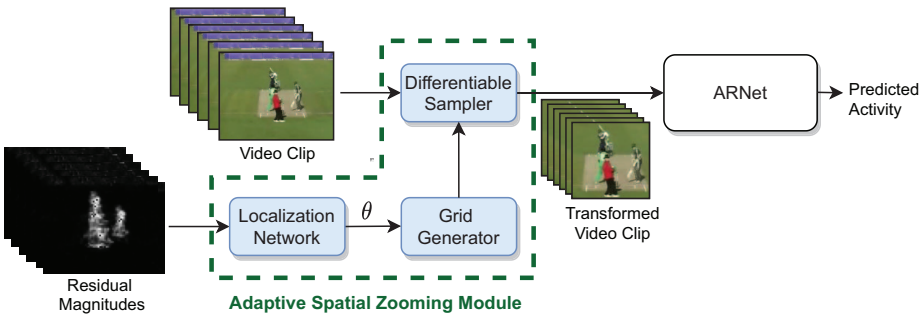

Figure 1. Schematic depiction of our proposed adaptive spatial zooming module for action recognition prediction. The adaptive spatial zooming module is delineated with the green dashed line which consists of a localization network, grid generator, and differentiable sampler. ARNet can be any 3D CNN AR model that processes video-clip data.

2D per-frame processing, 6) We introduce motion-related maps from compressed video domain to guide our adaptive learning process.

Spatial Attention for Action Recognition. Earlier methods on spatial attention for AR, include memory-networks that work with feature-derived positional heatmaps [28], non-local networks [37], and attentional-pooling networks [9]. More recently, transformer networks were introduced [3], where video frames are parsed into a sequence of non-overlapping patches, and frame features with self-attention maps are learnt. Our work bears some similarity to Video-LSTM [18]. The latter learns spatial attention saliency maps sequentially, using weak classification labels, and then thresholds the maps to define action localization boxes. This work has relied on motion information to aid defining saliency, but in contrary to that we do not use optical flow, and instead make use of sparser and computationally much more efficient compressed domain modalities. To our knowledge no works so far have used motion-clues from compressed domain to learn attention. Further, we learn to regress CROIs, and avoid using any heuristics in CROI definition (i.e. thresholding).

Lastly, conceptually, our work may resemble a previous work on attention learning in image classification from [24], where interesting glimpses of the input data are processed at high-resolution, likewise achieving high apparent input resolution. However, our system is end-to-end differentiable, we process single CROIs per input frames, the method is applied to action recognition in videos and also uses different underlying modalities to guide attention learning of CROIs.

3. Adaptive Spatial Zooming Using Compressed Domain Modalities

Here, we explain the details of our proposed method which uses the compressed video domain modalities (more specifically, motion vectors and residuals) to learn spatial attention for optimal region zooming for the action recognition task. Following our experimental evidence (as detailed in Experiments), demonstrating the superiority of residuals over other modalities, raw or compressed, for learning the spatial attention, we introduce and explain the problem setup and methods mostly on the example of residuals. We first explain the problem setup in section 3.1. Then, we discuss the details of our proposed methodology in section 3.2.

3.1. Problem Setup

As introduced in Section 2, in this work we use raw video data, complemented with compressed domain modalities. We assume that both the decoded RGB images and the relevant compressed domain information, i.e. motion vectors and residuals, are available from a compressed video for action recognition. Note that since we use the complete collection of decoded RGB images, we explicitly disregard I-frames from compressed video domain, but use them implicitly as those are a subset of decoded RGB images. Specifically, we consider a compressed video consisting of K frames, so by decoding this video, we can retrieve K RGB images, i.e., $\mathbf{V} = [\mathbf{I}_1, \mathbf{I}_2, \cdots, \mathbf{I}_K] \in \mathbb{R}^{K\times C\times H\times W}$, where $\mathbf{I}_i \in \mathbb{R}^{C\times H\times W}$ represents the RGB image of the i-th frame, as well as K motion vectors $\mathbf{M} = [\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_K] \in \mathbb{R}^{K\times C\times H\times W}$, and K motion residuals $\mathbf{R} = [\mathbf{R}_1, \mathbf{R}_2, \cdots, \mathbf{R}_K] \in \mathbb{R}^{K\times C\times H\times W}$, where $\mathbf{R}_i \in \mathbb{R}^{C\times H\times W}$ represents the residual of the i-th frame, while $\mathbf{M}_i \in \mathbb{R}^{C\times H\times W}$ denotes the motion vector associated with the i-th frame. H, W, and C denote the frame height, width, and number of color channels, respectively. Given this setup, our aim is to improve the action recognition task with the aid of the extra compressed domain information. In the following section, we present the details of our proposed method.

3.2. Learning Spatial Attention Aided with Compressed Domain Modalities

The system pipeline consists of four modules as depicted in Fig. 1, namely the localization network, the grid genera-

tor, the sampler (comprising together the proposed adaptive attention module), and the action recognition network (ARNet). We focus on the adaptive attention module, and in the following, we first give a brief overview of its architecture, and then explain the details of each of its components, as well as the steps taken to preprocess the input data. The ARNet can in general be any 3D CNN AR network that models video-clips. Our method can also be extended to 2D CNN AR methods.

3.2.1 Architecture Overview

Our proposed adaptive spatial attention method for action recognition prediction uses a light-weight localization network to predict the spatial transformation parameters θ given the compressed domain modalities (motion vectors or residuals) corresponding to the video clip. Then, the video clip is transformed using the sampling grid created by the grid generator and a differentiable sampler [13]. Finally, the transformed video clip is passed to the action recognition network (ARNet) for the final activity prediction.

Given an action recognition dataset $\mathcal{D} = \{(\mathbf{V}_i, y_i)\}_{i=0}^{P-1}$ with P videos, where each video $\mathbf{V}_i \in \mathcal{V} = \{\mathbf{V}_0, \mathbf{V}_1, \cdots, \mathbf{V}_{P-1}\}$ is labeled from C predefined action classes $y_i \in \mathcal{Y} = \{0, 1, \cdots, C-1\}$, our goal is to learn an adaptive spatial attention of the video-clip to improve the action recognition classification task $\mathcal{F}_{AR} : \mathcal{V} \rightarrow \mathcal{Y}$. To this end, we propose incorporating a light-weight network to efficiently learn the spatial transformation of a video clip using the compressed domain data as depicted in Fig. 1.

Let $\mathbf{V} \in \mathbb{R}^{K\times C\times H\times W}$ be the RGB video clip, $\mathbf{R} \in \mathbb{R}^{K\times C\times H\times W}$ be the residual clip (similarly, $\mathbf{M} \in \mathbb{R}^{K\times C\times H\times W}$ a motion vector clip), and $\mathcal{F}_{AR}$ be the action classifier network. First, the individual residuals in $\mathbf{R}$ (likewise, motion vectors $\mathbf{M}$) are accumulated, normalized, and resized resulting in $\mathbf{R}_A \in \mathbb{R}^{C\times H^a\times W^a}$ ($\mathbf{M}_A \in \mathbb{M}^{C\times H^a\times W^a}$, for motion vectors), where H^a and W^a are the height and width of the resized accumulated residuals, respectively. Then, the processed residual clip is passed to a localization network (LNet) to predict optimal transformation parameters θ which are later used to perform the spatial transformation. We work with differentiable affine transformations. In order to allow for an end-to-end training process, the transformation should be implemented as a differentiable module. To this end, first a sampling grid is generated using the estimated transformation parameters θ [13]. Then, given the input clip $\mathbf{V}$ and the generated sampling grid, the transformed output $\mathbf{V}' \in \mathbb{R}^{K\times C\times H'\times W'}$ is computed using a differentiable sampler (a bilinear sampler in our case). Finally, the transformed video clip is used as the input to the action classifier network $\mathcal{F}_{AR}$. In the following section, we present the details of our proposed method.

3.2.2 Architecture Details

Our proposed adaptive attention method consists of two major modules as depicted in Fig. 1: the adaptive spatial zooming module comprising the localization network, the grid generator, and the differentiable sampler, and the ARNet module. As aforesaid, ARNet can in general be any 3D CNN AR model that processes video-clip data. In the following, we focus on the spatial zooming module and present the details of each component, as well as the steps taken to preprocess the input data.

Preprocessing of Compressed Video Clip. To estimate the transformation parameters θ, the localization network uses the residual clip or the motion vector clip. We explain the preprocessing steps using residuals. The preprocessing of motion vectors is identical in the most part, with the only difference in the number of channels between the two modalities. Let $\mathbf{R} \in \mathbb{R}^{K\times C\times H\times W}$ denote a residual clip, with the individual residuals $\mathbf{R}_i \in \mathbb{R}^{C\times H\times W}$ for $i \in \{0, 1, \cdots, K-1\}$. Here, the preprocessing steps of the residual clip is explained.

First, K residuals associated with the RGB clip are extracted from the compressed video data [39]. The original residuals coming from the compressed domain are noisy and they represent only the difference between two consecutive frames (motion compensated). To capture longer term changes and to increase the signal-to-noise ratio, we accumulate the original residuals, inside their respective GoPs, following the approach from [39]. Next, we create a single-channel residual magnitude image $\hat{\mathbf{R}}_i \in \mathbb{R}^{H\times W}$ for each individual 3-channel residual $\mathbf{R}_i = [\mathbf{R}_{i,0}, \mathbf{R}_{i,1}, \mathbf{R}_{i,2}] \in \mathbb{R}^{3\times H\times W}$, from clip $\mathbf{R}$, using the following equation:

$$\hat{\mathbf{R}}_i = \sqrt{\sum_{c=0}^{2} \mathbf{R}_{i,c}^2}, \quad \forall i \in \{0, 1, \cdots, K-1\}, \qquad (1)$$

where $\mathbf{R}_{i,c}$ is the c-th channel of the i-th residual $\mathbf{R}_i$, the square and square root operations are done element-wise, and the summation operation is done channel-wise. Next, $\hat{\mathbf{R}}_i$'s are normalized with respect to their corresponding maximum value and downscaled to $H^a \times W^a$, where $H^a = H/S$ and $W^a = W/S$ for some $S > 1$. Finally, the 2D residuals $\hat{\mathbf{R}}_i$'s for $i \in \{1, 2, \cdots, K\}$ are concatenated to form a 3D residual clip $\mathbf{R}_A \in \mathbb{R}^{K\times H^a\times W^a}$ which is used as the input of the localization network as explained in the next section.

Localization Network (LNet). To allow for an end-to-end training of both the adaptive attention and the action recognition networks using the standard back-propagation algorithm, all the involved modules presented in Fig. 1 must be differentiable. To meet this requirement, we train a localization network (LNet) to learn the differentiable transformation parameters θ and then use these learned parameters

to perform a differentiable spatial transformation. We use $\mathcal{F}_{LNet}(\cdot)$, which is a light weight convolutional neural network (CNN), to estimate θ given the pre-processed residual input $\mathbf{R}_A \in \mathbb{R}^{K\times H^a\times W^a}$ as

$$\theta = \mathcal{F}_{LNet}(\mathbf{R}_A; \theta_{LNet}), \tag{2}$$

where θ_{LNet} stands for the trainable parameters of LNet. The sparse nature of residuals allows us to use a light-weight LNet. To further decrease the computational cost at this step, we downscale the accumulated residuals as described in the previous section.

Grid Generator and Differentiable Sampler. The differentiable transformation is implemented using the grid generator and the differentiable sampler, proposed in Spatial-Transformer Networks [13]. As shown in Fig. 1, the differentiable transformation parameters θ are passed to the grid generator to create a sampling grid. Then, this sampling grid along with the video clip are passed to the differentiable sampler to perform the transformation operation. We apply the same transformation to all the K frames $\mathbf{I}_i$'s of a clip $\mathbf{V} = [\mathbf{I}_1, \mathbf{I}_2, \cdots, \mathbf{I}_K] \in \mathbb{R}^{K\times C\times H\times W}$. Here, we explain the differentiable transformation procedure for a single frame $\mathbf{I}$.

To create a transformed frame $\mathbf{I}_c$ of size $C \times H' \times W'$ from the original frame $\mathbf{I}$ of size $C \times H \times W$, we first define the output pixel grid for $\mathbf{I}_c$ of the desired size ($C\times H'\times W'$). The sample points in the original frame $\mathbf{I}$ are calculated from the output pixel grid using the regressed transformation parameters θ. The coordinates of both the original and output image are normalized by the respective image width and height, and lie inside $[-1, 1]$. A differentiable bilinear sampler is then employed to populate the output pixel grid from the interpolated sample points [13], making the spatial transformation module differentiable and allowing for end-to-end differentiable training pipeline.

The same transformation is applied to all the frames of the input video clip and the results are concatenated to form the transformed clip $\mathbf{V}_c = [\mathbf{I}_{\mathbf{c}1}, \mathbf{I}_{\mathbf{c}2}, \cdots, \mathbf{I}_{\mathbf{c}K}] \in \mathbb{R}^{K\times C\times H'\times W'}$, where $\mathbf{I}_{\mathbf{c}i} \in \mathbb{R}^{C\times H'\times W'}$ represents the i-th transformed frame for $i \in \{1, 2, \cdots, K\}$. In a mathematical form, this procedure can be summarized as follows

$$\mathbf{V}_c = \mathcal{F}_{SM}(\mathbf{V}, \theta), \tag{3}$$

where $\mathcal{F}_{SM}$ stands for the sampler module. Finally, $\mathbf{V}_c$ is passed to the action recognition network $\mathcal{F}_{ARNet}$ with trainable parameters θ_{ARNet} to predict the activity $\hat{y}$ as follows

$$\hat{y} = \mathcal{F}_{ARNet}(\mathbf{V}_c; \theta_{ARNet}). \tag{4}$$

Combining equations (2) to (4) results in

$$\hat{y} = \mathcal{F}_{ARNet}(\mathcal{F}_{SM}(\mathbf{V}, \mathcal{F}_{LNet}(\mathbf{R}_A; \theta_{LNet})); \theta_{ARNet}). \tag{5}$$

Because all the modules in (5) are differentiable, it is possible to train parameters of both LNet (θ_{LNet}) and ARNet (θ_{ARNet}) in an end-to-end manner using the standard back-propagation algorithm and a loss function $\mathcal{L}(y, \hat{y})$. In the next section, we present the experimental and ablation results of our proposed method.

4. Experiments

In this section, we first explain our experimental setups including datasets, network architectures, as well as training and inference setups. Then, we investigate the effect of plugging our proposed adaptive zooming into three widely used clip-level 3D ARNets. We also do ablation studies to compare our approach with other alternatives. Finally, we present samples to visually compare our proposed adaptive transformation method with that of the widely used central crop.

4.1. Experimental Setup

Datasets. We conduct our experiments on four public datasets, namely UCF101 [32], HMDB51[17], Kinetics-Temporal [27], and Kinetics-Static [27]. UCF101 and HMDB51 are well studied datasets in action recognition. UCF101 contains 13320 videos from 101 action classes of realistic scenarios with camera movement and cluttered background. HMDB51 includes 6766 videos from 51 action classes. For each class label, there is at least 101 clips collected from movie scenes and the web. Kinetics-Temporal is a subset of Kinetics-400 [15] that contains 18096 videos for training and 1588 videos for validation of the 32 classes with significant temporal information. Kinetics-Static contains 20904 videos for training and 1593 videos for validation of the 32 classes with comparatively fewer motions. For all the experiments, we used MPEG-4 Part-2 encoded videos.

Network Architectures. In all experiments, we use the EfficientNet-B0 architecture [33] pretrained on the ImageNet dataset [5] for the LNet. We consider three different architectures R3D-18 [34], R(2+1)D [35], and X3D [8] pretrained on Kinetics-400 [15] with no optical flow input for the ARNet architecture. R3D18 contains 3D convolutions following space-time pool and a fully connected layer. R(2+1)D architecture is built upon R3D by replacing the 3D convolution layers with (2+1)D convolutions. X3D, on the other hand, is an expansion of X2D through progressive algorithm in temporal, spatial, and channel dimensions.

Training and Inference. We use the following setup for the baseline. We set the number of frames of each clip to 16. We use one randomly selected clip at training, and 10 uniformly selected clips at inference. For R3D/R(2+1)D and X3D networks, we rescale the input clips to 128×171 and 256×342 frames, respectively. At training, we randomly crop input clip into 112×112 and 224×224 for

Table 1. The effect of plugging in our proposed adaptive zooming module to AR models. Here, top-1 and top-5 accuracy of the validation sets of three splits of HMDB51 are reported. Models with AZ outperform those without AZ in all splits and criteria.

Model	Top-1				Top-5			
Dataset Split	1	2	3	Average	1	2	3	Average
R3D	54.94	55.69	53.99	54.87	85.15	83.46	84.77	84.46
R3D + AZ	**59.78**	**59.67**	**60.13**	**59.86**	**88.36**	**88.69**	**86.73**	**87.93**
R(2+1)D	65.73	65.75	65.82	65.77	91.43	90.98	90.13	90.85
R(2+1)D + AZ	**68.48**	**66.93**	**68.10**	**67.84**	**93.33**	**92.81**	**92.29**	**92.81**
X3D	71.16	69.48	71.83	70.82	94.05	93.46	92.88	93.46
X3D + AZ	**72.73**	**71.37**	**72.68**	**72.26**	**94.44**	**93.66**	**93.07**	**93.72**

Table 2. The effect of plugging in our proposed adaptive zooming module to AR models. Here, top-1 and top-5 accuracy of the validation sets of three splits of UCF101 are reported. Models with AZ outperform those without AZ in all splits and criteria.

Model	Top-1				Top-5			
Dataset Split	1	2	3	Average	1	2	3	Average
R3D	85.75	85.65	85.17	85.52	97.78	97.64	97.62	97.68
R3D + AZ	**87.42**	**87.60**	**86.99**	**87.34**	**98.28**	**98.37**	**98.78**	**98.48**
R(2+1)D	92.76	93.41	92.91	93.03	99.21	99.20	99.40	99.27
R(2+1)D + AZ	**93.95**	**94.27**	**93.07**	**93.76**	**99.37**	**99.38**	**99.46**	**99.40**
X3D	92.81	94.13	93.02	93.32	99.37	99.44	99.16	99.32
X3D + AZ	**93.79**	**94.32**	**93.34**	**93.82**	**99.47**	**99.63**	**99.49**	**99.53**

R3D/R(2+1)D and X3D networks, respectively. At inference, we use center crop of size 112×112 and 224×224 for R3D/R(2+1)D and X3D networks, respectively [35, 8]. We use 8 GPUs and mini-batch of 256 (32 samples per GPU). Throughout training, we set the initial learning rate to 0.01 and 0.001 for UCF101/HMDB51 and Kinetics-Temporal/Static, respectively. We reduce the learning rate by the factor of 0.1 whenever there is no improvement in loss value. We use synchronous distributed Stochastic Gradient Descent (SGD) with the momentum of 0.1 as the optimizer.

For our proposed adaptive spatial attention method, we follow Coviar approach to extract and accumulate residuals [40]. We downscale the residuals with a factor of 4 ($S = 4$) resulting in 64×86 residual frames. The rest setup is the same as that of the baseline except for the cropping part where we use our adaptive method to perform the cropping at both training and inference time.

4.2. Main results

We compare our results with three action recognition architectures in presence or absence of Adaptive Zooming (AZ) mechanism on three public action recognition datasets. First, as it is shown in Table 1, our adaptive transformation approach outperforms all three models in HMDB51. R3D performance is improved the most when the adaptive transformation is in place in all three splits of HMDB51 dataset. Second, Table 2 also shows that the improvement is consistent for all splits of UCF101 dataset.

Table 3. The effect of plugging in our proposed adaptive zooming module to AR models. Here, top-1 and top-5 accuracy of the validation sets of temporal and static splits of Kinetics-32 are reported. Models with AZ outperform those without AZ in all splits and criteria.

Model	Top-1		Top-5	
Dataset Split	Temporal	Static	Temporal	Static
R3D	60.52	74.95	92.51	93.60
R3D + AZ	**65.93**	**81.04**	**94.77**	**97.05**
R(2+1)D	76.76	87.82	97.48	98.31
R(2+1)D + AZ	**77.46**	**88.58**	**97.82**	**98.74**
X3D	61.40	69.81	92.13	90.02
X3D + AZ	**73.05**	**84.81**	**96.60**	**97.36**

Third, we compare our models on temporal and static Kinetics-32. As it is illustrated in Table 3, AZ is still able to improve the performance whether the scene is static or not. This suggest that adaptive transformation take more than only the movement of objects or camera into account.

Lastly, we report the computational cost of adaptive AZ in terms of Multiply-Accumulate operations (MACs). The computation cost of R3D, R(2+1)D, and X3D is 8.33, 40.71, and 4.97, respectively. By adding AZ, the computation cost increases 0.006 GMac which is negligible compared to the computational cost of each model. Therefore, the LNet computation cost is negligible.

Table 4. Adaptive scale comparison with and without shear and translation.

Inference cropping	Top-1	Top-5
Adaptive all	64.75	91.37
Adaptive scale and translation	67.10	92.81
Adaptive scale and shear	63.77	90.84
Adaptive scale	**67.95**	**92.81**

4.3. Ablation studies

We perform four ablation studies on HMDB51 dataset by constraining our base model to R(2+1)D. We first investigate what θ parameter contributes more in action recognition accuracy. Then, we discuss the effect of data augmentation in our experiments. Third, we consider different input types to our LNet network. Finally, we compare our model with simple random and center crop.

Transformation Parameters. The transformation matrix θ is an affine transformation matrix that consists of six parameters. We employ these parameters to create an affine transformation. The transformations are either translation, shear, scale, or any combination of them. As clarified in Section 3, the scale transformation is the basic block of our work. Therefore, we compare translation and shear along with scale transformation. Particularly, we consider translation and scale, shear and scale, all three, and scale only experiments. If a transformation is not targeted in the experiment, it means that its corresponding parameters are set to zero. As it is shown in Table 4, scale adaptive transformation results in the best performance compared to other adaptive methods. In the following, we study the effect of data augmentation on the performance.

Data Augmentation. Considering Table 4, our ablation suggests optimizing the scale parameters only. However, we still find it useful to use the additional parameters from θ to allow for a stronger data augmentation for the ARNet. More specifically, we apply data augmentation in the form of shear and translation to the transformation matrix θ. As a result, an augmented transformation is given to ARNet input. Table 5 shows the superior accuracy performance of this type of data augmentation.

Table 5. Comparison of model with adaptive scale in two scenarios. First row: without data augmentation. Second row: with data augmentation.

Data Augmentation	Top-1	Top-5
No	67.95	92.81
Yes	**68.48**	**99.33**

LNet Input. We compare different types of inputs to LNet network including RGB frame, motion vectors and residuals as inputs. Table 6 illustrates that residual inputs outperform other types of inputs.

Table 6. Comparison of our model (adaptive scale crop with data augmentation) with different input types to LNet.

LNet input	Top-1	Top-5
MV	68.15	93.13
RGB	65.99	92.41
Res	**68.48**	**99.33**

Table 7. Comparison of random and center crop with adaptive crop. The input to the LNet is residuals in our adaptive model.

Inference cropping	Top-1	Top-5
Random	64.55	90.65
Center (92.5%)	65.27	91.43
Center (87.5%)	65.73	91.43
Center (80%)	63.44	90.78
Center (75%)	58.93	88.36
Center (60%)	9.81	27.93
Center (mean adaptive ratio)	66.32	92.41
Adaptive	**68.48**	**99.33**

Adaptive Crop vs. Non-Adaptive Crop. One question that has remained unanswered is whether non-adaptive cropping with no overhead cost are alternatives to the adaptive cropping. In Table 7, we present evaluation results obtained by center crop with central fraction $\alpha \in [0.6, 0.75, 0.8, 0.875, .925]$ and random crop where the randomly selected area is set to cover anything between 80% to 100% of the input. Table 7 shows that although both random and center crops are done with no computation cost, they underperform our scale adaptive approach with residual inputs. We also apply center cropping whose ratio is obtained by averaging the box cropping ratio of our adaptive method. In this case, the results show a superior performance of our adaptive method as well.

Adaptive vs. Center Cropping Visualization. Finally, we present the visualization of our proposed adaptive transformation versus center cropping in figure 2. We use R(2+1)D model and the HMDB51 dataset for this visualization. As the presented samples of this figure depict, our proposed method results in tighter classification-regions-of-interests.

5. Conclusions

We have proposed an efficient methodology and a pluggable module for transforming input video-clips with adaptive spatial zooming transformation, in 3D CNN AR modeling, using compressed video modalities. Our experimental work demonstrated direct benefits of the proposed approach. One limitation of this work is related to it not being applicable to raw videos for which compressed representa-

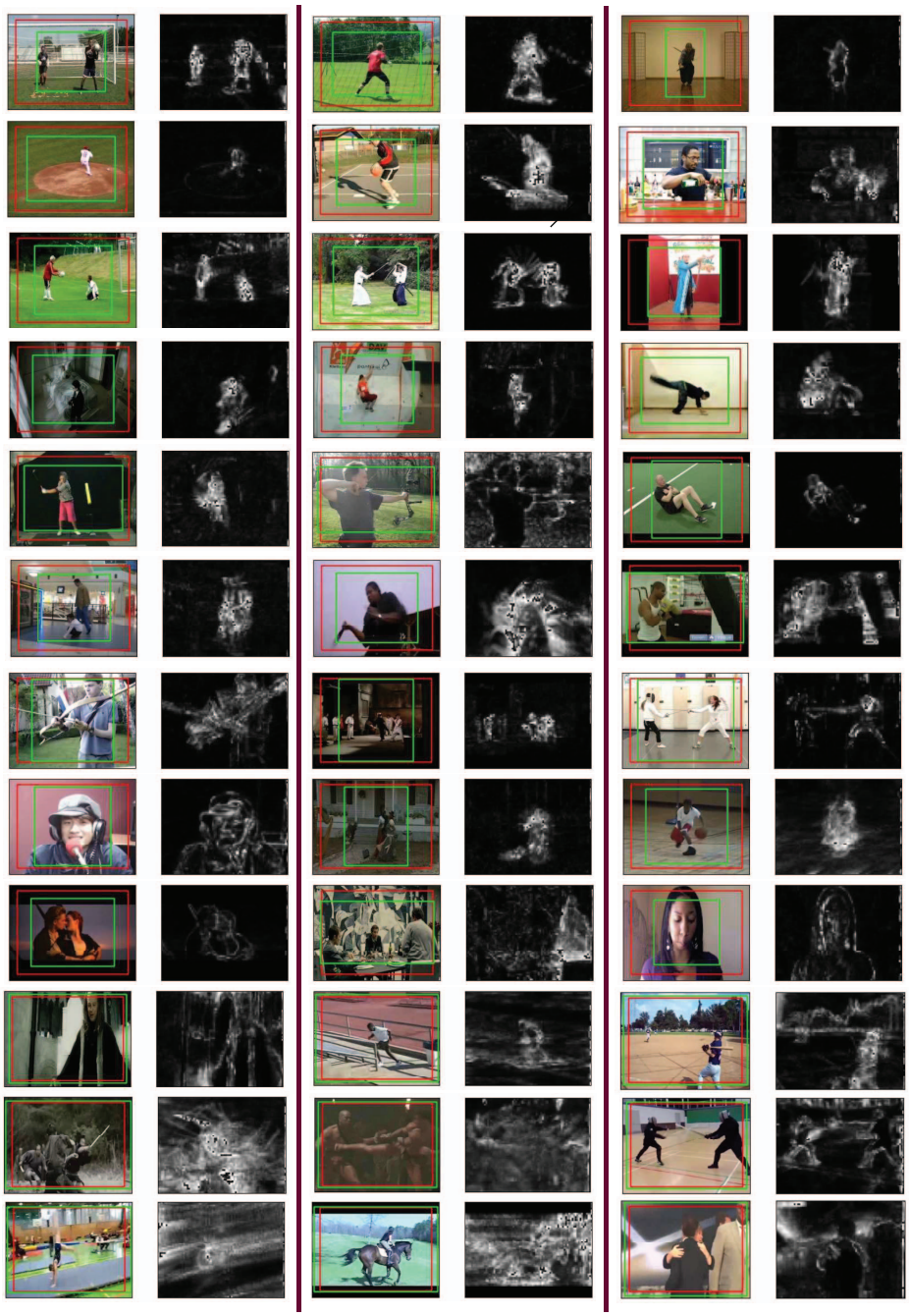

Figure 2. The visualization our proposed adaptive transformation versus center cropping. On the left of each RGB-residual pair of images is the middle frame of a 16-frame video clip, and on the right of each pair is the channel-wise mean of all the 16-frame residual clip. The red and green bounding boxes represent the center crop and our proposed adaptive transformation, respectively.

tion is not available. However, in that case, we can rely on RGB modality to guide attention learning, albeit with more modest positive effects, as discussed in ablation studies. Another potential pitfall is related to static videos associated to action categories with more important semantic than motion features. Nonetheless, our experimentation on carefully chosen static and temporal dataset representatives suggest there are still benefits in using motion clues, and that we systematically improve the baseline accuracy in such cases as well. The future work will look into generalizing and adapting the approach to 2D AR.

References

[1] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *CoRR*, abs/1412.7755, 2015.

[2] Marlene Behrmann and Craig Haimson. The cognitive neuroscience of visual attention. *Current Opinion in Neurobiology*, 9:158–163, 1999.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 813–824. PMLR, 2021.

[4] Haoyuan Cao, Shining Yu, and Jiashi Feng. Compressed video action recognition with refined motion vector. *arXiv preprint arXiv:1910.02533*, 2019.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, 2015.

[7] John Duncan. Converging levels of analysis in the cognitive neuroscience of visual attention. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 353 1373:1307–17, 1998.

[8] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[9] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *NIPS*, 2017.

[10] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[11] Hezhen Hu, Wengang Zhou, Xingze Li, Ning Yan, and Houqiang Li. MV2Flow: Learning motion representation for fast compressed video action recognition. *ACM Trans. on Multimedia Computing, Communications, and Applications (TOMM)*, 16(3s), Jan. 2021.

[12] Yuqi Huo, Xiaoli Xu, Yao Lu, Yulei Niu, Mingyu Ding, Zhiwu Lu, Tao Xiang, and Ji-rong Wen. Lightweight action recognition in compressed videos. In *Proc. of the European Conf. on Computer Vision*, pages 337–352, 2020.

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[14] 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*, 35(1):221–231, 2013.

[15] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[16] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6231–6241, 2019.

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

[18] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees G.M. Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018.

[19] Zhaoyang Liu, Donghao Luo, Yabiao Wang ansd Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *The Thirty-Fourth (AAAI) Conference on Artificial Intelligence, (AAAI) 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, (IAAI) 2020, The Tenth (AAAI) Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 11669–11676. (AAAI) Press, 2020.

[20] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13688–13698, 2021.

[21] Chih-Yao Ma, Min-Hung Chen, Zsolt Kira, and Ghassan Al-Regib. Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *Signal Process. Image Commun.*, 71:76–87, 2019.

[22] Lili Meng, Bo Zhao, B. Chang, Gao Huang, Wei Sun, Fred Tung, and Leonid Sigal. Interpretable spatio-temporal attention for video action recognition. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1513–1522, 2019.

[23] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and Rogério Schmidt Feris. Ar-net: Adaptive frame resolution for efficient action recognition. In *ECCV*, 2020.

[24] Volodymyr Mnih, Nicolas Manfred Otto Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *NIPS*, 2014.

[25] Rameswar Panda, Chun-Fu Richard Chen, Quanfu Fan, Ximeng Sun, Kate Saenko, Aude Oliva, and Rogerio Feris. Adamml: Adaptive multi-modal learning for efficient video recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7556–7565, 2021.

[26] Ronald A. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7:17 – 42, 2000.

[27] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. pages 535–544, 2021.

[28] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *CoRR*, abs/1511.04119, 2015.

[29] Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. DMC-net: Generating discriminative motion cues for fast compressed video action recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[32] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[33] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[34] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[35] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[36] 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 Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, volume 9912 of *Lecture Notes in Computer Science*, pages 20–36. Springer, 2016.

[37] X. Wang, Ross B. Girshick, Abhinav Kumar Gupta, and Kaiming He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[38] Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. Adaptive focus for efficient video recognition. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 16229–16238. IEEE, 2021.

[39] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6026–6035, June 2018.

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

[41] Zuxuan Wu, Caiming Xiong, Yu-Gang Jiang, and Larry S. Davis. Liteeval: A coarse-to-fine framework for resource efficient video recognition. *Int. J. Comput. Vis.*, 129:2965–2977, 2019.

[42] Ke Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[43] Yuan Zhi, Zhan Tong, Limin Wang, and Gangshan Wu. Mgsampler: An explainable sampling strategy for video action recognition. In *2021 IEEE/CVF International Conference on Computer Vision, (ICCV) 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 1493–1502. IEEE, 2021.