Temporal Aggregate Representations for Long-Range Video Understanding

Fadime Sener[1,2], Dipika Singhania[2], and Angela Yao[2]

[1] University of Bonn, Germany
[2] National University of Singapore
{sener@cs.uni-bonn.de},{dipika16,ayao}@comp.nus.edu.sg

Abstract. Future prediction, especially in long-range videos, requires reasoning from current and past observations. In this work, we address questions of temporal extent, scaling, and level of semantic abstraction with a flexible multi-granular temporal aggregation framework. We show that it is possible to achieve state of the art in both next action and dense anticipation with simple techniques such as max-pooling and attention. To demonstrate the anticipation capabilities of our model, we conduct experiments on Breakfast, 50Salads, and EPIC-Kitchens datasets, where we achieve state-of-the-art results. With minimal modifications, our model can also be extended for video segmentation and action recognition.

Keywords: action anticipation, temporal aggregation

1 Introduction

We tackle long-range video understanding, specifically anticipating not-yet observed but upcoming actions. When developing intelligent systems, one needs not only to recognize what is *currently* taking place – but also predict what will happen *next*. Anticipating human actions is essential for applications such as smart surveillance, autonomous driving, and assistive robotics.

While action anticipation is a niche (albeit rapidly growing) area, the key issues that arise are germane to long-range video understanding as a whole. How should temporal or sequential relationships be modelled? What temporal extent of information and context needs to be processed? At what temporal scale should they be derived, and how much semantic abstraction is required? The answers to these questions are not only entangled with each other but also depend very much on the videos being analyzed. Here, one needs to distinguish between clipped actions, e.g. of UCF101 [32], versus the multiple actions in long video streams, e.g. of the Breakfast [18]. In the former, the actions and video clips are on the order of a few seconds, while in the latter, it is several minutes. As such, temporal modelling is usually not necessary for simple action recognition [13], but more relevant for understanding complex activities [28, 30].

Temporal models that are built into the architecture [6, 8, 12, 29] are generally favoured because they allow frameworks to be learned end-to-end. However,

this means that the architecture also dictates the temporal extent that can be accounted for. This tends to be short, either due to difficulties in memory retention or model size. As a result, the context for anticipation can only be drawn from a limited extent of recent observations, usually on the order of seconds [19, 38, 26]. This, in turn, limits the temporal horizon and granularity of the prediction.

One way to ease the computational burden, especially under longer temporal extents, is to use higher-level but more compact feature abstractions, e.g. by using detected objects, people [42] or sub-activity labels [1, 15] based on the outputs of video segmentation algorithms [29]. Such an approach places a heavy load on the initial task of segmentation and is often difficult to train end-to-end. Furthermore, since labelling and segmenting actions from video are difficult tasks, their errors may propagate onwards when anticipating future actions.

Motivated by these questions of temporal modelling, extent, scaling, and level of semantic abstraction, we propose a general framework for encoding long-range video. We aim for flexibility in frame input, i.e. ranging from low-level visual features to high-level semantic labels, and the ability to meaningfully integrate recent observations with long-range context in a computationally efficient way. To do so, we split video streams into snippets of equal length and max-pool the frame features within the snippets. We then create ensembles of multi-scale feature representations that are aggregated bottom-up based on scaling and temporal extent. Temporal aggregation [16] is a form of summarization used in database systems. Our framework is loosely analogous as it summarizes the past observations through aggregation, so we name it "temporal aggregates". We summarize our main contributions as follows:

- We propose a simple and flexible single-stage framework of multi-scale temporal aggregates for videos by relating recent to long-range observations.
- Our representations can be applied to several video understanding tasks; in addition to action anticipation, it can be used for recognition and segmentation with minimal modifications and is able to achieve competitive results.
- Our model has minimal constraints regarding the type of anticipation (dense or next action), type of the dataset (instructional or daily activities), and type of input features (visual features or frame-level labels).
- We conduct experiments on Breakfast [18], 50Salads [34] and EPIC-Kitchens [5].

2   Related Works

Action recognition has advanced significantly with deep networks in recent years. Notable works include two steam networks [31, 40], 3D convolutional networks [36, 3], and RNNs [7, 44]. These methods have been designed to encode clips of a few seconds and are typically applied to the classification of *trimmed* videos containing a single action [32, 14]. In our paper, we work with long *untrimmed* sequences of complex activities. Such long videos are not simply a composition of independent short actions, as the actions are related to each other with sequence dynamics. Various models for complex activity understanding have been addressed before [6, 8, 30]; these approaches are designed to work

on instructional videos by explicitly modelling their sequence dynamics. These models are not flexible enough to be extended to daily activities with loose orderings. Also, when only partial observations are provided, e.g. for anticipation, these models cannot be trained in a single stage.

Action anticipation aims to forecast actions before they occur. Prior works in immediate anticipation were initially limited to movement primitives like *reaching* [17] or interactions such as *hugging* [38]. [24] presents a model for predicting both the next action and its starting position. [5] presents a large daily activities dataset, along with a challenge for anticipating the next action one second before occurrence. [26] proposes next action anticipation from recent observations. Recently, [10] proposed using an LSTM to summarize the past and another LSTM for future prediction. These works assume near-past information, whereas we make use of long-range past.

Dense anticipation predicts actions multiple steps into the future. Previous methods [1, 15] to date require having already segmented temporal observations. Different than these, our model can perform dense anticipation in a single stage without any pre-segmented nor labelled inputs.

The **role of motion and temporal dynamics** has been well-explored for video understanding, though the focus has been on short clips [22, 3, 13]. Some works use longer-term temporal contexts, though still in short videos [21, 35]. Recently, Wu *et al.* [42] proposed integrating long-term features with 3D CNNs in short videos and showed the importance of temporal context for action recognition. Our model is similar in spirit to [42] in that we couple the recent with the long-range past using attention. One key difference is that we work with ensembles of multiple scalings and granularities, whereas [42] work at a single frame-level granularity. As such, we can handle long videos up to tens of minutes, while they are only able to work on short videos. Recently, Feichtenhofer *et al.* [9] proposed SlowFast networks, which, similar to our model, encode time-wise multi-scale representations. These approaches are limited to short videos and cannot be extended to minutes-long videos due to computational constraints.

3   Representations

We begin by introducing the representations, which are inputs to the building blocks of our framework, see Fig. 1. We had two rationales when designing our network. First, we relate recent to long-range observations, since some past actions directly determine future actions. Second, to represent recent and long-range past at various granularities, we pool snippets over multiple scales.

3.1   Pooling

For a video of length T, we denote the feature representation of a single video frame indexed at time t as $f_t \in \mathbb{R}^D, 1 \leq t \leq T$. f_t can be derived from low-level features, such as IDT [39] or I3D [3], or high-level abstractions, such as
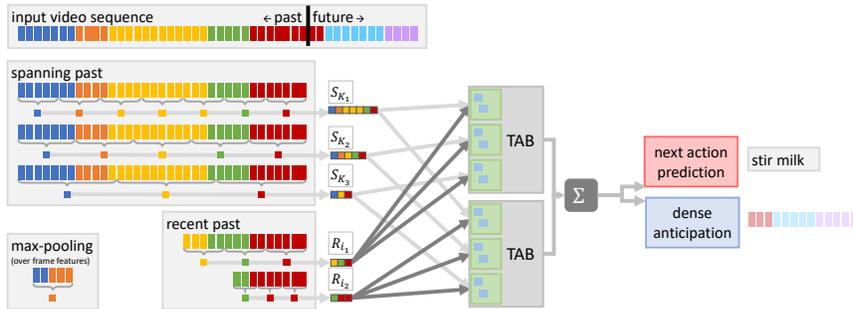
Fig. 1. Model overview: In this example we use 3 scales for computing the "spanning past" features $\mathbf{S}_{K_1}, \mathbf{S}_{K_2}, \mathbf{S}_{K_3}$, and 2 starting points to compute the "recent past" features, $\mathbf{R}_{i_1}, \mathbf{R}_{i_2}$, by max-pooling over the frame features in each snippet. Each recent snippet is coupled with all the spanning snippets in our Temporal Aggregation Block (TAB). An ensemble of TAB outputs is used for dense or next action anticipation.

sub-activity labels derived from segmentation algorithms. To reduce computational load, we work at a snippet-level and define a snippet feature $\mathbf{F}_{ij;K}$ as the concatenation of max-pooled features from K snippets, where snippets are partitioned consecutively from frames i to j:

$$\mathbf{F}_{ij;K} = [F_{i,i+k}, F_{i+k+1,i+2k}, ..., F_{j-k+1,j}], \text{ where}$$
$$(F_{p,q})_d = \max_{p \le t \le q} \{f_t\}_d, \ 1 \le d \le D, \ k = (j-i)/K. \tag{1}$$

Here, $F_{p,q}$ indicates the maximum over each dimension d of the frame features in a given snippet between frames p and q, though it can be substituted with other alternatives. In the literature, methods representing snippets or segments of frames range from simple sampling and pooling strategies to more complex representations such as learned pooling [23] and LSTMs [27]. Especially for long snippets, it is often assumed that a learned representation is necessary [11, 20], though their effectiveness over simple pooling is still controversial [40]. The learning of novel temporal pooling approaches goes beyond the scope of this work and is an orthogonal line of development. We verify established methods (see Sec. 5.2) and find that a simple max-pooling is surprisingly effective and sufficient.

3.2   Recent vs. Spanning Representations

Based on different start and end frames i and j and number of snippets K, we define two types of snippet features: *"recent"* features $\{\mathcal{R}\}$ from recent observations, and *"spanning"* features $\{\mathcal{S}\}$ drawn from the entire video. The recent snippets cover a couple of seconds (or up to a minute, depending on the temporal granularity) before the current time point while spanning snippets refer to the long-range past and may last up to ten minutes. For *"recent"* snippets, the end frame j is fixed to the current time point t, and the number of snippets is
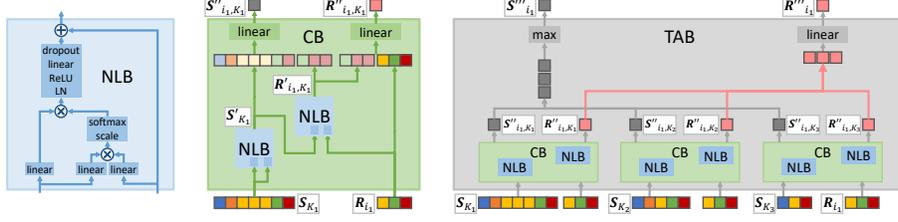
Fig. 2. Model components: Non-Local Blocks (NLB) compute interactions between two representations via attention (Sec. 4.1). Two such NLBs are combined in a Coupling Block (CB), which calculates the attention-reweighted spanning and recent representations (Sec. 4.2). We couple each recent with all spanning representations via individual CBs and combine their outputs in a Temporal Aggregation Block (TAB) (Sec. 4.3). The outputs of multiple such TABs are combined to perform anticipation, Fig. 1.

fixed to K_R. The recent snippet features $\mathcal{R}$ can be defined as a feature bank of snippet features with different start frames i, i.e.

$$\mathcal{R} = \{\mathbf{F}_{i_1 t; K_R}, \mathbf{F}_{i_2 t; K_R}, ..., \mathbf{F}_{i_R t; K_R}\} = \{\mathbf{R}_{i_1}, \mathbf{R}_{i_2}, ..., \mathbf{R}_{i_R}\}, \tag{2}$$

where $\mathbf{R}_i \in \mathbb{R}^{D \times K_R}$ is a shorthand to denote $\mathbf{F}_{i t; K_R}$, since endpoint t and number of snippets K_R are fixed. In Fig. 1 we use two starting points to compute the "recent" features and represent each with $K_R = 3$ snippets (■■■ & ■■■).

For *"spanning"* snippets, i and j are fixed to the start of the video and current time point, i.e. $i = 0, j = t$. Spanning snippet features $\mathcal{S}$ are defined as a feature bank of snippet features with varying number of snippets K, i.e.

$$\mathcal{S} = \{\mathbf{F}_{0 t; K_1}, \mathbf{F}_{0 t; K_2}, ..., \mathbf{F}_{0 t; K_S}\} = \{\mathbf{S}_{K_1}, \mathbf{S}_{K_2}, ..., \mathbf{S}_{K_S}\}, \tag{3}$$

where $\mathbf{S}_K \in \mathbb{R}^{D \times K}$ is a shorthand for $\mathbf{F}_{0 t; K}$. In Fig. 1 we use three scales to compute the "spanning" features with $K = \{7, 5, 3\}$ (■■■■■■■,■■■■■ &■■■).

Key to both types of representations is the ensemble of snippet features from multiple scales. We achieve this by varying the number of snippets K for the spanning past. For the recent past, it is sufficient to keep the number of snippets K_R fixed, and vary only the start point i, due to redundancy between $\mathcal{R}$ and $\mathcal{S}$ for the snippets that overlap. For our experiments, we work with snippets ranging from seconds to several minutes.

4 Framework

In Fig. 2 we present an overview of the components used in our framework, which we build in a bottom-up manner, starting with the recent and spanning features $\mathcal{R}$ and $\mathcal{S}$, which are coupled with non-local blocks (NLB) (Sec. 4.1) within coupling blocks (CB) (Sec. 4.2). The outputs of the CBs from different scales are then aggregated inside temporal aggregation blocks (TAB) (Sec. 4.3). Outputs of different TABs can then be chained together for either next action anticipation or dense anticipation (Secs. 5.3, 5.5).

4.1   Non-Local Blocks (NLB)

We apply non-local operations to capture relationships amongst the spanning snippets and between spanning and recent snippets. Non-local blocks [41] are a flexible way to relate features independently from their temporal distance and thus capture long-range dependencies. We use the modified non-local block from [42], which adds layer normalization [2] and dropout [33] to the original one in [40]. Fig. 2 (left) visualizes the architecture of the block, the operation of which we denote as $NLB(\cdot, \cdot)$.

4.2   Coupling Block (CB)

Based on the NLB, we define attention-reweighted spanning and recent outputs:

$$\mathbf{S}'_K = NLB(\mathbf{S}_K, \mathbf{S}_K) \quad \text{and} \quad \mathbf{R}'_{i,K} = NLB(\mathbf{S}'_K, \mathbf{R}_i). \tag{4}$$

$\mathbf{R}'_{i,K}$ is coupled with either $\mathbf{R}_i$ or $\mathbf{S}'_K$ via concatenation and a linear layer. This results in the fixed-length representations $\mathbf{R}''_{i,K}$ and $\mathbf{S}''_{i,K}$, where i is the starting point of the recent snippet and K is the scale of the spanning snippet.

4.3   Temporal Aggregation Block (TAB)

The final representation for recent and spanning past is computed by aggregating outputs from multiple CBs. For the same recent starting point i, we concatenate $\mathbf{R}''_{i,K_1}, ..., \mathbf{R}''_{i,K_S}$ for all spanning scales and pass the concatenation through a linear layer to compute $\mathbf{R}'''_i$. The final spanning representation $\mathbf{S}'''_i$ is a max over all $\mathbf{S}''_{i,K_1}, ..., \mathbf{S}''_{i,K_S}$. We empirically find that taking the max outperforms other alternatives like linear layers and/or concatenation for the spanning past (Sec. 5.2). TAB outputs, by varying recent starting points $\{i\}$ and scales of spanning snippets $\{K\}$, are multi-granular video representations that aggregate and encode both the recent and long-range past. We name these **temporal aggregate representations**. Fig.1 shows an example with 2 recent starting points and 3 spanning scales. These representations are generic and can be applied in various video understanding tasks (see Sec. 4.4) from long streams of video.

4.4   Prediction Model

Classification: For single-label classification tasks such as next action anticipation, temporal aggregate representations can be used directly with a classification layer (linear + softmax). A cross-entropy loss based on ground truth labels Y can be applied to the predictions $\hat{Y}_i$, where Y is either the current action label for recognition, or the next action label for next action prediction (see Fig. 3).

When the individual actions compose a complex activity (e.g. "take bowl" and "pour milk" as part of "making cereal" in Breakfast [18]), we can add an additional loss based on the complex activity label Z. Predicting Z as an auxiliary task helps with anticipation. For this we concatenate $\mathbf{S}'''_{i_1}, \ldots, \mathbf{S}'''_{i_R}$ from
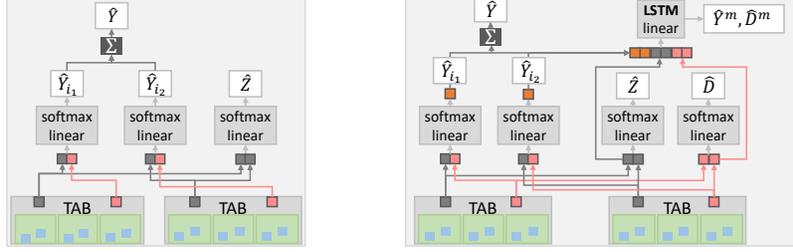
Fig. 3. Prediction models for classification (left) and sequence prediction (right).

all TABs and pass them through a classification layer to obtain $\hat{Z}$. The overall loss is a sum of the cross entropies over the action and complex activity labels:

$$\mathcal{L}_{\text{cl}} = \mathcal{L}_{\text{comp}} + \mathcal{L}_{\text{action}} = -\sum_{n=1}^{N_Z} Z_n \log(\hat{Z})_n - \sum_{r=1}^{R} \sum_{n=1}^{N_Y} Y_n \log(\hat{Y}_{i_r})_n, \qquad (5)$$

where i_r is one of the R recent starting points, and N_Y and N_Z are the total number of actions and complex activity classes respectively. During inference, the predicted scores are summed for a final prediction, i.e. $\hat{Y} = \max_n(\sum_{r=1}^{R} \hat{Y}_{i_r})_n$.

We frame sequence segmentation as a classification task and predict frame-level action labels of complex activities. Multiple sliding windows with fixed start and end times are generated and then classified into actions using Eq.5.

Sequence prediction: The dense anticipation task predicts frame-wise actions of the entire future sequence. Previously, [1] predicted future segment labels via classification and regressed the durations. We opt to estimate both via classification. The sequence duration is discretized into N_D intervals and represented as one-hot encodings $D \in \{0,1\}^{N_D}$. For dense predictions, we perform multi-step estimates. We first estimate the current action and complex activity label, as per Eq. 6. The current duration D is then estimated via a classification layer applied to the concatenation of recent temporal aggregates $\mathbf{R}'''_{i_1}, ..., \mathbf{R}'''_{i_R}$.

For future actions, we concatenate all recent and spanning temporal aggregates $\mathbf{R}'''_{i_1}, ..., \mathbf{R}'''_{i_R}$ and $\mathbf{S}'''_{i_1}, ..., \mathbf{S}'''_{i_R}$ and the classification layer outputs $\hat{Y}_{i_1}, ..., \hat{Y}_{i_R}$, and pass the concatenation through a linear layer before feeding the output to a one-layer LSTM. The LSTM predicts at each step m an action vector $\hat{Y}^m$ and a duration vector $\hat{D}^m$ (see Fig. 3). The dense anticipation loss is a sum of the cross-entropies over the current action, its duration, future actions and durations, and task labels respectively:

$$\mathcal{L}_{\text{dense}} = \mathcal{L}_{\text{cl}} - \sum_{n=1}^{N_D} D_n \log(\hat{D})_n - \frac{1}{M} \sum_{m=1}^{M} \left(\sum_{n=1}^{N_Y} Y_n^m \log(\hat{Y}^m)_n + \sum_{n=1}^{N_D} D_n^m \log(\hat{D}^m)_n \right)$$
$$(6)$$

During inference we sum the predicted scores (post soft-max) for all starting points i_r to predict the current action as $\max_n(\sum_{r=1}^{R} \hat{Y}_{i_r})_n$. The LSTM is then applied recurrently to predict subsequent actions and durations.

Dataset	video duration median, mean $\pm$std	# classes	# segments	$\{i\}$(in seconds)	K_R	$\{K\}$
Breakfast(@15fps)	15.1s, 26.6s $\pm$36.8	48	11.3K	$\{t-10, t-20, t-30\}$	5	$\{10, 15, 20\}$
50Salads(@30fps)	29.7s, 38.4s $\pm$31.5	17	0.9K	$\{t-5, t-10, t-15\}$	5	$\{5, 10, 15\}$
EPIC(@60fps)	1.9s, 3.7s $\pm$ 5.6	2513	36.6K	$\{t-1.6, t-1.2, t-0.8, t-0.4\}$	2	$\{2, 3, 5\}$

Table 1. Dataset details and our respective model parameters.

5   Experiments

5.1   Datasets and Features

We experiment on Breakfast [18], 50Salads [34] and EPIC-Kitchens [5]. The sequences in each dataset reflect realistic durations and orderings of actions, which is crucial for real-world deployment of anticipation models. Relevant datasets statistics are given in Table 1. One notable difference between these datasets is the label granularity; it is very fine-grained for EPIC, hence their 2513 action classes, versus the coarser 48 and 17 actions of Breakfast and 50Salads. As a result, the median action segment duration is 8-16x shorter.

Feature-wise, we use pre-computed Fisher vectors [1] and I3D [3] for Breakfast, Fisher vectors for 50Salads, and appearance, optical flow and object-based features provided by [10] for EPIC. Results for Breakfast and 50Salads are averaged over the predefined 4 and 5 splits respectively. Since 50Salads has only a single complex activity (making salad) we omit complex activity prediction for it. For EPIC, we report results on the test set. Evaluation measures are class accuracy (Acc.) for next action prediction and mean over classes [1] for dense prediction. We report Top-1 and Top-5 accuracies to be consistent with [26, 10].

Hyper-parameters for spanning $\{K\}$, recent scales K_R and recent starting points $\{i\}$ are given in Table 1. We cross validated the parameters on different splits of 50Salads and Breakfast; on EPIC, we select parameters with the validation set [10].

5.2   Component validation

We verify each component's utility via a series of ablation studies summarized in Table 2. As our main motivation was to develop a representation for anticipation in long video streams, we validate on Breakfast for next action anticipation. Our full model gets a performance of 40.1% accuracy averaged over actions.

Video Representation: Several short-term feature representations have been proposed for video, e.g. 3D convolutions [36], or combining CNNs and RNNs for sequences [44, 7]. For long video streams, it becomes difficult to work with all the raw features. Selecting representative features can be as simple as sub-sampling the frames [9, 43], or pooling [40], to more complex RNNs [44]. Current findings in the literature are not in agreement. Some propose learned strategies [25, 20], while others advocate pooling [40]. Our experiments align with the latter, showing that max-pooling is superior to both sampling (+8%) and the GRU (+2.2%) or bi-directional LSTM [4] (+1.4%). The performance of GRU and

Pooling type	frame sampling	GRU	BiLSTM	mean-pooling	max-pooling
Acc.	32.1	37.9	38.7	36.6	**40.1**

Influence of	Changes in components	Acc. (Drop)
Non-Local Blocks (NLB)	replace all NLBs with concatenation + linear layer	33.7 (6.4%)
Coupling Blocks (CB)	only couple the S_K and S_K in CBs	35.1 (5.0%)
	only couple the R_i and R_i in CBs	34.2 (5.9%)
	replace CBs with concatenation + linear layer	33.4 (6.7%)
Temporal Aggregation Blocks (TAB)	a single CB is used in TABs	38.0 (2.1%)
	three CBs are used in a single TAB	37.7 (2.4%)
	a single a CB is used without any TABs	32.1 (8.0%)

Recent & Spanning Repr.

(a)

starting points i	$i_1 = t - 10$	$i_2 = t - 20$	$i_3 = t - 30$	$i_4 = 0$	{$\mathbf{i_1}, \mathbf{i_2}, \mathbf{i_3}$}		
Acc.	36.9	37.7	37.2	35.1	40.1		

(b)

spanning scales K	{5}	{10}	{15}	{20}	{10, 15}	{10,15,20}	{5,10,15,20}
Acc.	37.4	38.0	37.5	37.4	39.0	**40.1**	40.2

(c)

recent scales K_R	1	3	**5**	10
Acc.	38.7	39.5	**40.1**	38.6

Table 2. Ablations on the influence of different model components.

BiLSTM are comparable to average-pooling, but require much longer training and inference time. For us, max-pooling works better than average pooling; this contradicts the findings of [40]. We attribute this to the fact that we pool over minutes-long snippets and it is likely that mean- smooths away salient features that are otherwise preserved by max-pooling. We conducted a similar ablations on EPIC, where we observed a 1.3% increase with max- over mean-pooling.

Recent and Spanning Representations: In our ablations, unless otherwise stated, an ensemble of 3 spanning scales $K = \{10, 15, 20\}$ and 3 recent starting points $i = \{t - 10, t - 20, t - 30\}$ are used. Table 2 **(a)** compares single starting points for the recent snippet features versus an ensemble. With a single starting point, points too near to and too far from the current time decrease the performance. The worst individual result is with $i_4 = 0$, i.e. using the entire sequence; the peak is at $i_2 = t - 20$, though an ensemble is still best. In Table 2 **(b)**, we show the influence of spanning snippet scales. These scales determine the temporal snippet granularity; individually, results are not significantly different across the scales, but as we begin to aggregate an ensemble, the results improve. The ensemble with 4 scales is best but only marginally better than 3, at the expense of a larger network, so we choose $K = \{10, 15, 20\}$. In Table 2 **(c)**, we show the influence of recent snippet scales, we find $K_R = 5$ performs best.

Coupling Blocks: Previous studies on simple video understanding have shown the benefits of using features from both the recent and long-range past [21, 42]. A naïve way to use both is to simply concatenate, though combining the two in a learned way, e.g. via attention, is superior (+6.4%). To incorporate attention, we apply NLBs [41], which is an adaptation of the attention mechanism that is popularly used in machine translation. When we replace our CBs with concatenation and a linear layer, there is a drop of 6.7%. When we do not use coupling but separately pass the R_i and S_K through concatenation and a linear layer, there is a drop of 7.5%. We find also that coupling the recent R_i and long range S_K information is critical. Coupling only recent information (-5.9%) does

Method	Input	Segmentation Method and Feature	Breakfast	50Salads
[38]	FC7 features	-	8.1	6.2
[26]	R(2+1)D	-	32.3	
RNN [1]	segmentation	[29], Fisher	**30.1**	30.1
CNN [1]	segmentation	[29], Fisher	27.0	29.8
ours no Z	Fisher	-	29.2	**31.6**
ours	Fisher	-	29.7	
ours	I3D	-	40.1	40.7
ours	segmentation	ours, I3D	43.1	
ours	segmentation + I3D	ours, I3D	47.0	
ours	frame GT	-	64.7	63.8
ours	frame GT + I3D	-	63.1	

Table 3. Next action anticipation comparisons on Breakfast and 50Salads, given different frame inputs (GT action labels , Fisher vectors , I3D features).

not keep sufficient context, whereas coupling only long-range past (-5%) does not leave sufficient representation for the more relevant recent aspects.

Temporal Aggregation Blocks (TAB) are the most critical component. Omitting them and classifying a single CB's outputs significantly decreases accuracy (-8%). The strength of the TAB comes from using ensembles of coupling blocks as input (single, -2.1%) and using the TABs in an ensemble (single, -2.4%).

Additional ablations: When we omit the auxiliary complex activity prediction, i.e. removing the Z term from Eq. 6 ("no Z"), we observe a slight performance drop of 1.1%. In our model we max pool over all $\mathbf{S}''_{i,K_1}, ..., \mathbf{S}''_{i,K_S}$ in our TABs. When we replace the max-pooling with concatenation + linear, we reach an accuracy of 37.4. We also try to disentangle the ensemble effect from the use of multi-granular representations. When we fix the spanning past scales K to $\{15, 15, 15\}$ and all the starting points to $i = t - 20$, we observe a drop of 1.2% in accuracy which indicates the importance of our multi-scale representation.

5.3   Anticipation on Procedural Activities - Breakfast & 50 Salads

Next Action Anticipation predicts the action that occurs 1 second from the current time t. We compare to the state of the art in Table 3 with two types of frame inputs: spatio-temporal features (Fisher vectors or I3D) and frame-wise action labels (either from ground truth or via a separate segmentation algorithm) on Breakfast. Compared to previous methods using only visual features as input, we outperform CNN (FC7) features [38] and spatio-temporal features R(2+1)D [26] by a large margin (+32.3% and +8.1%). While the inputs are different, R(2+1)D features were shown to be comparable to I3D features [37]. Since [26] uses only recent observations, we conclude that incorporating the spanning past into the prediction model is essential.

Our method degrades when we replace I3D with the weaker Fisher vectors (40.1% vs 29.7%). Nevertheless, this result is competitive with methods using action labels [1] (30.1% with RNN) derived from segmentation algorithms [29] using Fisher vectors as input. For fair comparison, we report a variant without the complex activity prediction ("no Z"), which has a slight performance drop (-0.5%). If we use action labels as inputs instead of visual features, our performance

|	Breakfast								50salads							
Obs.	20%				30%				20%				30%			
Pred.	10%	20%	30%	50%	10%	20%	30%	50%	10%	20%	30%	50%	10%	20%	30%	50%
A	**Labels** (GT)								**Labels** (GT)							
RNN[1]	60.4	50.4	45.3	40.4	61.5	50.3	45.0	41.8	42.3	31.2	25.2	16.8	44.2	29.5	20.0	10.4
CNN[1]	58.0	49.1	44.0	39.3	60.3	50.1	45.2	40.5	36.1	27.6	21.4	15.5	37.4	24.8	20.8	14.1
Ke[15]	64.5	**56.3**	**50.2**	**44.0**	66.0	55.9	**49.1**	**44.2**	45.1	33.2	27.6	17.3	**46.4**	**34.8**	**25.2**	13.8
ours	**65.5**	55.5	46.8	40.1	**67.4**	**56.1**	47.4	41.5	**47.2**	**34.6**	**30.5**	**19.1**	44.8	32.7	23.5	**15.3**
B	**Features** (Fisher)								**Features** (Fisher)							
CNN[1]	12.8	11.6	11.2	10.3	17.7	16.9	15.5	14.1								
ours	**15.6**	**13.1**	**12.1**	**11.1**	**19.5**	**17.0**	**15.6**	**15.1**	25.5	19.9	18.2	15.1	30.6	22.5	19.1	11.2
C	**Labels** (Fisher + [29] (Acc. 36.8/42.9))								**Labels** (Fisher + [29] (Acc. 66.8/66.7))							
RNN[1]	18.1	**17.2**	15.9	15.8	21.6	20.0	19.7	19.2	30.1	25.4	18.7	13.5	30.8	17.2	14.8	9.8
CNN[1]	17.9	16.4	15.4	14.5	22.4	20.1	19.7	18.8	21.2	19.0	16.0	9.9	29.1	20.1	17.5	10.9
Ke[15]	18.4	**17.2**	16.4	**15.8**	22.8	**20.4**	19.6	**19.8**	32.5	**27.6**	21.3	**16.0**	**35.1**	**27.1**	22.1	15.6
ours	**18.8**	16.9	**16.5**	15.4	**23.0**	20.0	**19.9**	18.6	**32.7**	26.3	**21.9**	15.6	32.3	25.5	**22.7**	**17.1**
	Concatenate B and C								Concatenate B and C							
ours	**25.0**	**21.9**	**20.5**	**18.1**	**23.0**	**20.5**	19.8	**19.8**	**34.7**	25.9	**23.7**	15.7	34.5	26.1	19.0	15.5
D	**Features** (I3D)															
ours	24.2	21.1	20.0	18.1	30.4	26.3	23.8	21.2								
E	**Labels** (I3D + our seg. (Acc. 54.7/57.8))															
ours	**37.4**	31.2	30.0	26.1	39.5	34.1	31.0	**27.9**								
	Concatenate D and E															
ours	37.1	**31.8**	**30.1**	**27.1**	**39.8**	**34.2**	**31.9**	27.8								

Table 4. Dense anticipation mean over classes on Breakfast and 50salads, given different frame inputs (GT action labels , Fisher vectors , I3D features).

improves from 40.1% to 43.1%; merging labels and visual features gives another 4% boost to 47%. In this experiment we use segmentation results from our own framework, (see Sec. 5.6). However, if we substitute ground truth instead of segmentation labels, there is still a 17% gap. This suggests that the quality of the segmentation matters. When the segmentation is very accurate, adding additional features does not help and actually slightly deteriorates results (see Table 3 "frame GT" vs. "frame GT + I3D").

In Table 3, we also report results for 50Salads. Using Fisher vectors we both outperform the state of the art by 1.8% and the baseline with CNN features [38] by 25.4%. Using I3D features improves the accuracy by 9.1% over Fisher vectors.

Dense Anticipation predicts frame-wise actions; accuracies are given for specific portions of the remaining video (Pred.) after observing a given percentage of the past (Obs.). We refer the reader to the supplementary for visual results. Competing methods [1] and [15] have two stages; they first apply temporal video segmentation and then use outputs [29], i.e. frame-wise action labels, as inputs for anticipation. We experiment with both action labels and visual features.

For Breakfast (Table 4, left), when using GT frame labels, we outperform the others, for shorter prediction horizons. For 50Salads (Table 4, right), we outperform the state of the art for the observed 20%, and our predictions are more accurate on long-range anticipation (Pred. 50%). We outperform [1] when we use visual features as input (B Features (Fisher)). When using the segmentations (from [29], which has a frame-wise temporal segmentation accuracy of 36.8% and 42.9% for the observed 20% and 30% of video respectively), we are comparable to state of the art [15]. We further merge visual features with action labels for dense anticipation. With Fisher vectors and the frame labels obtained
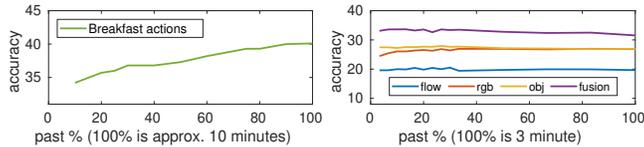
Fig. 4. Effect of spanning scope on instructional vs. daily activities. For EPIC we report Top-5 Acc. on the validation set with rgb, flow and object features and late fusion.

from [29], we observe a huge performance increase in performance compared to only using the frame labels (up to +7%) in Breakfast. In 50Salads, this increase is not significant nor consistent. This may be due to the better performing segmentation algorithm on 50Salads (frame-wise accuracy of 66.8% and 66.7% for 20% and 30% observed respectively). We observe further improvements on Breakfast once we substitute Fisher vectors with I3D features and segmentations from our own framework (I3D + ours seg.). Similar to next action anticipation, performance drops when using only visual features as input (I3D is better than Fisher vectors). When using I3D features and the frame label outputs of our segmentation method, we obtain our model's best performance, with a slight increase over using only frame label outputs.

5.4 How much spanning past is necessary?

We vary the duration of spanning snippets (Eq. 3) with start time i as fractions of the current time t; $i = 0$ corresponds to the full sequence, i.e. 100% of the spanning past, while $i = t$ corresponds to none, i.e. using only recent snippets since the end points j remain fixed at t. Using the entire past is best for Breakfast (Fig. 4 left). Interestingly, this effect is not observed on EPIC (Fig. 4 right). Though we see a small gain by 1.2% until 40% past for the appearance features (rgb), beyond this, performance saturates. We believe this has to do with the fine granularity of labels in EPIC; given that the median action duration is only 1.9s, one could observe as many as 16 actions in 30 seconds. Given that the dataset has only 28.5K samples split over 2513 action classes, we speculate that the model cannot learn all the variants of long-range relationships beyond 30 seconds. Therefore, increasing the scope of the spanning past does not further increase the performance. Based on experiments on the validation set, we set the spanning scope to 6 seconds for EPIC for the rest of the paper.

5.5 Recognition and Anticipation on Daily Activities - EPIC

The **anticipation** task of EPIC requires anticipating the future action $\tau_\alpha = 1s$ before it starts. For fair comparison to the state of the art [10] (denoted by "RU"), we directly use features (appearance, motion and object) provided by the authors. We train our model separately for each feature modality with the same hyper-parameters and fuse predictions from the different modalities by

		Action Anticipation						Action Recognition					
		Top-1 Accuracy%			Top-5 Accuracy%			Top-1 Accuracy%			Top-5 Accuracy%		
		Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
S1	[26]	30.7	16.5	9.7	76.2	42.7	25.4	-	-	-	-	-	-
	TSN [5]	31.8	16.2	6.0	76.6	42.2	28.2	48.2	36.7	20.5	84.1	62.3	39.8
	RU [10]	33.0	22.8	14.4	79.6	50.9	33.7	56.9	43.1	33.1	85.7	67.1	55.3
	LFB [42]	-	-	-	-	-	-	60.0	45.0	32.7	88.4	71.8	55.3
	[43]	-	-	-	-	-	-	65.7	46.4	35.9	89.5	71.7	57.8
	ours	31.4	22.6	16.4	75.2	47.2	**36.4**	63.2	49.5	41.3	87.3	70.0	63.5
	ours v+n	**37.9**	**24.1**	**16.6**	**79.7**	**54.0**	36.1	**66.7**	**49.6**	**41.6**	**90.1**	**77.0**	**64.1**
S2	[26]	28.4	12.4	7.2	69.8	32.2	19.3	-	-	-	-	-	-
	TSN [5]	25.3	10.4	2.4	68.3	29.5	6.6	39.4	22.7	10.9	74.3	45.7	25.3
	RU [10]	27.0	15.2	8.2	69.6	34.4	21.1	43.7	26.8	19.5	73.3	48.3	37.2
	LFB [42]	-	-	-	-	-	-	50.9	31.5	21.2	77.6	57.8	39.4
	[43]	-	-	-	-	-	-	**55.8**	32.7	24.0	**81.7**	58.9	43.2
	ours	27.5	**16.6**	10.0	66.8	32.8	23.4	52.0	31.5	26.2	76.8	52.7	45.7
	ours v+n	**29.5**	16.5	**10.1**	**70.1**	**37.8**	**23.4**	54.6	**33.5**	**27.0**	80.4	**61.0**	**46.4**

Table 5. Action anticipation and recognition on EPIC tests sets S1 and S2

voting. Note that for experiments on this dataset we do not use the entire past for computing our spanning snippet features (see Section 5.4). Results on hold-out test data of EPIC are given in Table 5 for seen kitchens (S1) with the same environments as the training data and unseen kitchens (S2) of held out environments. We outperform state of the art, RU [10], in the Top-1 and Top-5 action accuracies by 2% and 2.7% on S1 and by 1.8% and 2.3% on S2 using the same features suggesting superior temporal reasoning abilities of our model. When we add verb and noun classification to our model as auxiliary tasks to help with anticipation, "ours v+n", our performance improves for action and especially for noun and verb scores. For challenge results see supplementary.

For **recognition**, we classify pre-trimmed action segments. We adjust the scope of our spanning and recent snippets according to the action start and end times t_s and t_e. Spanning features are computed on a range of $[t_s - 6, t_e + 6]$; the first recent snippet scope is fixed to $[t_s, t_e]$ and the rest to $[t_s - 1, t_e + 1], [t_s - 2, t_e + 2]$ and $[t_s - 3, t_e + 3]$. Remaining hyper-parameters are kept the same. In Table 5, we compare to state of the art; we outperform all other methods including SlowFast networks with audio data [43] (+5.4% on S1, +2.2% on S2 for Top-1) and LFB [42], which also uses non-local blocks (+8.6% on S1, +5% on S2 for Top-1) and RU [10] by approximately +7% on both S1 and S2. Together with the anticipation results we conclude that our method generalizes to both anticipation and recognition tasks and is able to achieve state-of-the-art results on both, while [10] performs very well on anticipation but poorly on recognition.

5.6 Temporal Video Segmentation

We compare our performance against the state of the art, MS-TCN (I3D) [8], in Table 6 on Breakfast. We test our model with 2s and 5s windows. We report the frame-wise accuracy (Acc), segment-wise edit distance (Edit) and F1 scores at overlapping thresholds of 10%, 25% and 50%. In the example sequences, in the F1 scores and edit distances in Table 6, we observe more fragmentation in our segmentation for 2s than for 5s. However, for 2s, our model produces better accuracies, as the 5s windows are smoothing the predictions at action boundaries.

	F1@{10, 25, 50}			Edit	Acc.
MS-TCN (I3D) [8]	52.6	48.1	37.9	**61.7**	**66.3**
ours I3D 2s	52.3	46.5	34.8	51.3	65.3
ours I3D 5s	**59.2**	**53.9**	**39.5**	54.5	64.5
ours I3D GT.seg.	-	-	-	-	**75.9**

Table 6. Exemplary segmentation and comparisons on Breakfast.

Additionally we provide our model's upper bound, "ours I3D GT.seg.", for which we classify GT action segments instead of sliding windows. The results indicate that there is room for improvement, which we leave as future work. We show that we are able to easily adjust our method from its main application and already get close to the state of the art with slight modifications.

6   Discussion & Conclusion

This paper presented a temporal aggregate model for long-range video understanding. Our method computes recent and spanning representations pooled from snippets of video that are related via coupled attention mechanisms. Validating on three complex activity datasets, we show that temporal aggregates are either comparable or outperform the state of the art on three video understanding tasks: action anticipation, recognition and temporal video segmentation.

In developing our framework, we faced questions regarding temporal extent, scaling, and level of semantic abstraction. Our experiments show that max-pooling is a simple and efficient yet effective way of representing video snippets; this is the case even for snippets as long as two minutes. For learning temporal relationships in long video, attention mechanisms relating the present to long range context can successfully model and anticipate upcoming actions. The extent of context that is beneficial, however, may depend on the nature of activity (instructional vs. daily) and label granularity (coarse vs. fine) of the dataset.

We found significant advantages to using ensembles of multiple scales, both in recent and spanning snippets. Our aggregates model is flexible and can take as input either visual features or frame-wise action labels. We achieve competitive performance with either form of input, though our experiments confirm that higher levels of abstraction such as labels are more preferable for anticipation. Nevertheless, there is still a large gap between what can be anticipated with inputs from current segmentation algorithms in comparison to ground truth labels, leaving room for segmentation algorithms to improve.

References

1. Abu Farha, Y., Richard, A., Gall, J.: When will you do what?-anticipating temporal occurrences of activities. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5343–5352 (2018)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6299–6308 (2017)
4. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 670–680 (2017)
5. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Scaling egocentric vision: The dataset. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 753–771. Springer International Publishing, Cham (2018)
6. Ding, L., Xu, C.: Weakly-supervised action segmentation with iterative soft boundary assignment. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6508–6516 (2018)
7. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2625–2634 (2015)
8. Farha, Y.A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3575–3584 (2019)
9. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6202–6211 (2019)
10. Furnari, A., Farinella, G.M.: What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In: International Conference on Computer Vision (ICCV) (2019)
11. Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., Russell, B.: Actionvlad: Learning spatio-temporal aggregation for action classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 971–980 (2017)
12. Huang, D.A., Fei-Fei, L., Niebles, J.C.: Connectionist temporal modeling for weakly supervised action labeling. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 137–153. Springer International Publishing, Cham (2016)
13. Huang, D.A., Ramanathan, V., Mahajan, D., Torresani, L., Paluri, M., Fei-Fei, L., Carlos Niebles, J.: What makes a video a video: Analyzing temporal information in video understanding models and datasets. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7366–7375 (2018)
14. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)

15. Ke, Q., Fritz, M., Schiele, B.: Time-conditioned action anticipation in one shot. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
16. Kline, N., Snodgrass, R.T.: Computing temporal aggregates. In: Proceedings of the Eleventh International Conference on Data Engineering. pp. 222–231. IEEE (1995)
17. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. IEEE transactions on pattern analysis and machine intelligence (PAMI) **38**(1), 14–29 (2015)
18. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 780–787 (2014)
19. Lan, T., Chen, T.C., Savarese, S.: A hierarchical representation for future action prediction. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 689–704. Springer International Publishing, Cham (2014)
20. Lee, J., Natsev, A.P., Reade, W., Sukthankar, R., Toderici, G.: The 2nd youtube-8m large-scale video understanding challenge. In: Leal-Taixé, L., Roth, S. (eds.) Computer Vision – ECCV 2018 Workshops. pp. 193–205. Springer International Publishing, Cham (2019)
21. Li, F., Gan, C., Liu, X., Bian, Y., Long, X., Li, Y., Li, Z., Zhou, J., Wen, S.: Temporal modeling approaches for large-scale youtube-8m video understanding. arXiv preprint arXiv:1707.04555 (2017)
22. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 7083–7093 (2019)
23. Lin, R., Xiao, J., Fan, J.: Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. In: Leal-Taixé, L., Roth, S. (eds.) Computer Vision – ECCV 2018 Workshops. pp. 206–218. Springer International Publishing, Cham (2019)
24. Mahmud, T., Hasan, M., Roy-Chowdhury, A.K.: Joint prediction of activity labels and starting times in untrimmed videos. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5773–5782 (2017)
25. Miech, A., Laptev, I., Sivic, J.: Learnable pooling with context gating for video classification. arXiv preprint arXiv:1706.06905 (2017)
26. Miech, A., Laptev, I., Sivic, J., Wang, H., Torresani, L., Tran, D.: Leveraging the present to anticipate the future in videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 0–0 (2019)
27. Ostyakov, P., Logacheva, E., Suvorov, R., Aliev, V., Sterkin, G., Khomenko, O., Nikolenko, S.I.: Label denoising with large ensembles of heterogeneous neural networks. In: Leal-Taixé, L., Roth, S. (eds.) Computer Vision – ECCV 2018 Workshops. pp. 250–261. Springer International Publishing, Cham (2019)
28. Richard, A., Gall, J.: Temporal action detection using a statistical language model. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 3131–3140 (2016)
29. Richard, A., Kuehne, H., Gall, J.: Weakly supervised action learning with rnn based fine-to-coarse modeling. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 754–763 (2017)
30. Sener, F., Yao, A.: Unsupervised learning and segmentation of complex activities from video. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (2018)

31. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
32. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR **abs/1212.0402** (2012), http://arxiv.org/abs/1212.0402
33. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
34. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. pp. 729–738. ACM (2013)
35. Tang, Y., Zhang, X., Wang, J., Chen, S., Ma, L., Jiang, Y.G.: Non-local netvlad encoding for video classification. In: Leal-Taixé, L., Roth, S. (eds.) Computer Vision – ECCV 2018 Workshops. pp. 219–228. Springer International Publishing, Cham (2019)
36. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 4489–4497 (2015)
37. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6450–6459 (2018)
38. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating visual representations from unlabeled video. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 98–106 (2016)
39. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 3551–3558 (2013)
40. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 20–36. Springer International Publishing, Cham (2016)
41. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7794–7803 (2018)
42. Wu, C.Y., Feichtenhofer, C., Fan, H., He, K., Krähenbühl, P., Girshick, R.: Long-Term Feature Banks for Detailed Video Understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
43. Xiao, F., Lee, Y.J., Grauman, K., Malik, J., Feichtenhofer, C.: Audiovisual slowfast networks for video recognition. arXiv preprint arXiv:2001.08740 (2020)
44. Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4694–4702 (2015)