Deep multiple aggregation networks for action recognition

Ahmed Mazari, Hichem Sahbi

Deep Multiple Aggregation Networks for Action Recognition

Ahmed Mazari and Hichem Sahbi
Sorbonne University, CNRS, LIP6 F-75005, Paris, France

Abstract

Most of the current action recognition algorithms are based on deep networks which stack multiple convolutional, pooling and fully connected layers. While convolutional and fully connected operations have been widely studied in the literature, the design of pooling operations that handle action recognition, with different sources of temporal granularity in action categories, has comparatively received less attention, and existing solutions rely mainly on max or averaging operations. The latter are clearly powerless to fully exhibit the actual temporal granularity of action categories and thereby constitute a bottleneck in classification performances.

In this paper, we introduce a novel hierarchical pooling design that captures different levels of temporal granularity in action recognition. Our design principle is coarse-to-fine and achieved using a tree-structured network; as we traverse this network top-down, pooling operations are getting less invariant but timely more resolute and well localized. Learning the combination of operations in this network — which best fits a given ground-truth — is obtained by solving a constrained minimization problem whose solution corresponds to the distribution of weights that capture the contribution of each level (and thereby temporal granularity) in the global hierarchical pooling process. Besides being principled and well grounded, the proposed hierarchical pooling is also video-length and resolution agnostic. Extensive experiments conducted on the challenging UCF-101, HMDB-51 and JHMDB-21 databases corroborate all these statements.

Keywords. Multiple aggregation design 2-stream networks action recognition

1 Introduction

Action recognition is standing as one of the most challenging problems which consists in assigning one or multiple semantic categories to moving objects. This

task is difficult as scenes are usually acquired under extremely challenging conditions including cluttered background, viewpoint change, illumination variation, poor camera sensor quality and resolution. This affects the accuracy of multiple related applications such as scene understanding [34, 35, 36, 103], video surveillance [37, 38], video caption generation and retrieval [39, 41, 42, 43, 44, 45, 46, 47, 48] as well as human computer interaction and robotics [49, 50, 51, 52]. Most of the existing action recognition solutions are based on machine learning (ML) [9, 10, 11, 12, 13, 79]; their general recipe consists in learning functions that map video sequences into categories using widely used ML algorithms [32, 31, 63, 64, 108, 40]. Among the existing ML solutions those based on deep networks are currently witnessing a major interest [30, 23, 57, 58, 55, 56, 97, 8] but their success is tributary to the availability of large amount of labeled training data and also the appropriate choice of their architectures [2, 6, 7, 16, 14, 15, 29].

Successful architectures for action recognition include two-stream 2D/3D convolutional neural networks (CNNs) [1] operating on appearance and motion flows [2, 6], CNNs combined with Long Short-Term Memory (LSTM) networks [75] as well as 3D CNNs [7]. However, the effort in the design of these CNNs has focused essentially on optimizing their convolutional and fully connected layers while optimizing their pooling has comparatively received less attention. The difficulty in designing architectures with suitable pooling (a.k.a aggregation) operators, particularly on video sequences, stems from the eclectic properties of videos (namely their duration, temporal resolution and velocity of moving objects as well as the granularity of their action categories). Different pooling operators have been introduced in the literature and most of them are based on global measures including max and averaging [65, 66, 67], and more sophisticated ones [3, 4, 99] rely on visual saliency measures and attention which allow keeping only the most relevant information for the subsequent layers. These operators play a key role in reducing the dimensionality and the number of parameters of the learned representations, and thereby their dependency on large collections of labeled training data. While operators, such as global and max pooling, are well adapted to coarsely-grained actions, they turn out to be less advantageous on finely-grained categories (compared to other methods such as spectrograms) [2, 6, 7, 14, 15]. However, both averaging and spectrogram representations suffer from several drawbacks; average pooling built upon global statistical measures are time/duration agnostic but less discriminating *while* spectrogram-like methods are discriminant but time/duration aware and thereby highly sensitive to the acquisition conditions.

A more suitable pooling should gather the advantages of *averaging* and *spectrogram* representations while discarding their inconvenients. Following this goal, we consider in this paper a hierarchical aggregation scheme that describes moving scenes at multiple temporal granularities while also being resilient to their variable acquisition conditions. Top levels in this hierarchical aggregation provide orderless (invariant) but less discriminating representations (which capture coarse-grained action categories) *while* bottom levels correspond to fine-grained, timely resolute and order-sensitive representations. The design principle of our proposed solution is *coarse-to-fine* and allows us to capture a gradual

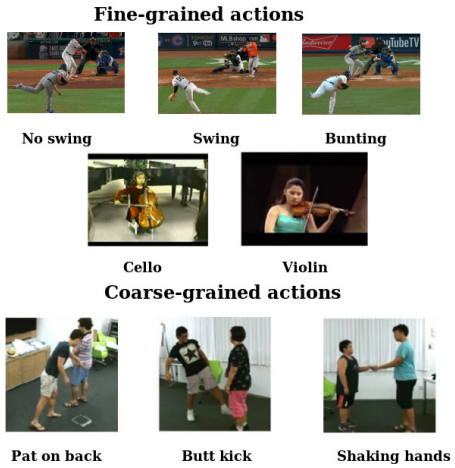


Figure 1: *Examples of fine and coarse-grained actions. The first row shows three action categories from the MLB-YouTube dataset [73]: "No swing", "Swing" and "Bunting" which are difficult to distinguish as they have very small differences. The second row shows two instrument playing actions from the UCF-101 dataset [19]: "cello" and "violin" which are also difficult to distinguish as their arm/hand locations and directions are similar. In contrast, the third row shows "Pat on back", "Butt kick" and "Shaking hand" actions (taken from NTU RGB+D dataset [74]) which are easier to distinguish.*

change of invariance and granularity; as we traverse the hierarchy top-down, our video representations are getting less invariant but timely more resolute and fine-grained. However, knowing a priori which combination of levels in this hierarchy is the most appropriate to capture the actual granularity of our video data is challenging, and learning this combination "end-to-end" is rather more appropriate.

Following this line, other related works [5, 24, 25, 26, 27, 28] try to model granularity of actions in videos by incorporating specific modules into CNNs beyond global average pooling and spectrograms. Two major categories of methods have been proposed; pyramidal methods and attention. In the first category, the method in [24] samples, from each video, frames as well as their associated optical flow components and adds a spatio-temporal pyramid module to CNN in order to capture hierarchical relationships between appearance and motion features. The approach in [25] stacks a temporal pyramid pooling layer on top of motion and appearance CNN streams in order to build fixed-length video representations. In [26], authors sample a set of frames by first splitting videos into segments and taking frames from each segment, and build a spatial pyramid to extract multi-scale appearance features from different convolutional layers. These features are then concatenated and fed to a three level temporal pyramid. The work in [27] samples video frames at different temporal resolu-

tions, and feeds them to a 3D CNN to extract their respective features followed by a temporal pyramid which down-samples and concatenates the resulting features. Authors in [95] propose the Interactive Aggregation Feature Pyramid Network (IA-FPN) which first aggregates 2D and 3D convolution features, and then fuse them at different resolutions. Finally, the method in [28] achieves frame sampling followed by a temporal pyramid pooling to build features at different pyramidal levels; the resulting features are afterwards fed to a temporal relational layer that groups these features at different scales.

In the second category of methods (i.e., attention-based), [100] propose two modules to boost 3D CNNs performances, based on temporal-channel correlation and bilinear pooling. Authors in [5] present a hierarchical bidirectional self-attention network to encode spatial-temporal information for actions. This attention network turns out to be effective in capturing both long-term temporal dependencies and spatial context information in videos. Subsequent work [94] presents a new architecture called self-attention pooling-based long-term temporal network (SP-LTN), which can learn long-term temporal representations (similarly to [101]) and aggregate those discriminative representations in an end-to-end manner. Authors in [96] propose two deep neural networks based on residual Fast-Slow Refined Highway and Global Atomic Spatial Attention to effectively detect and recognize actions. The work in [98] combines conditional random fields (CRFs) with self-attention in order to infer temporal and spatial dependencies. This combination benefits from the capability of CRFs in modeling dependencies, and self-attention in learning temporal evolution and spatial context in videos. In the particular scenario of skeleton-based recognition, [90] propose an efficient hierarchical self-attention network (HAN) for skeleton-based gesture recognition. A joint spatial and temporal self-attention module is used to aggregate, in a hierarchical way, joints-fingers-hands representations and dynamics prior to their classification.

The aforementioned pooling methods either have inherent limitations in capturing the dynamic of interacting parts in videos (such as global average pooling and spectrograms), or are computationally very overwhelming (such as attention-based models). While *more tractable and still effective methods* rely on hierarchical temporal aggregation schemes, none of them considers the issue of learning the best combination of levels in these temporal aggregation hierarchies, and this turns out to be highly effective as shown in the following sections and later in experiments.

In this paper, we introduce a novel scheme for action recognition based on Deep Multiple Aggregation Networks. Given a hierarchy of aggregation operations, the goal is to learn a combination of these operations that best fits a given action recognition ground-truth. We solve this problem by minimizing a constrained objective function whose parameters correspond to the distribution of weights through multiple aggregation levels; each weight captures the granularity of its level and its contribution in the global learned video representation. Beside handling aggregation at different levels, the particularity of our solution resides in its ability to handle variable length videos (without any up or down-

sampling) and thereby makes it possible to fully benefit from the whole frames in videos.

Considering all the aforementioned issues, the main contributions of this paper include

- A novel hierarchical aggregation block which models video sequences at *multiple levels of temporal granularity*. This block is learned "end-to-end" together with the parameters of a backbone convolutional network using both motion and appearance streams. A *reparametrization trick* is also introduced in order to implement equality and inequality constraints associated to the parameters of the aggregation block.

- An effective training procedure that allows handling videos with *varying* lengths without any *up-or-down* sampling, and thereby allows leveraging all the information in videos.

- An efficient training that allows *shuffling* gradients through multiple frames leading to high *speed-ups* in gradient estimation, and hence parameter updates, even when handling large video sequences.

- Extensive experiments on widely used databases show the validity of the proposed method and its efficiency.

The rest of this paper is organized as follows. First, we describe in Section 2 our hierarchical video representation based on motion and appearance streams. Then, we introduce in Section 3 our main contribution; a novel "end-to-end" two stream CNN training that aggregates and combine frame-level representations into temporal pyramids in order to achieve action recognition. Finally, we show in Section 4 the validity of these contributions through extensive experiments using standard and challenging video datasets including UCF-101, HMDB-51 and JHMDB-21. Finally, we conclude the paper while providing possible extensions for a future work.

2 Coarse-to-fine video representation

Let $\mathcal{S} = \{\mathcal{V}_i\}_{i=1}^{n}$ denote a collection of videos with each one being a sequence of frames $\mathcal{V}_i = \{f_{i,t}\}_{t=1}^{T_i}$ and let $\mathcal{C} = \{1, \ldots, C\}$ be a set of action categories (a.k.a classes). In order to describe the visual content of a given video $\mathcal{V}_i$, we rely on a two-stream process (see Fig. 2); the latter provides a complete description of appearance and motion that characterizes the spatio-temporal aspects of moving objects and their interactions. The output of the appearance stream (denoted as $\{\phi_a(f_{i,t})\}_{t=1}^{T_i} \subset \mathbb{R}^{2048}$) corresponds to an intermediate feature map taken from the deep residual network (ResNet-101; see Fig. 2) trained on ImageNet [23] and fine-tuned on UCF-101 [19]. The output of the motion stream (denoted as $\{\phi_m(f_{i,t})\}_{t=1}^{T_i} \subset \mathbb{R}^{2048}$) also corresponds to an intermediate feature map taken from the ResNet-101 network but trained on optical flow image pairs [71, 17]; these pairs correspond to the horizontal and the vertical displacement fields

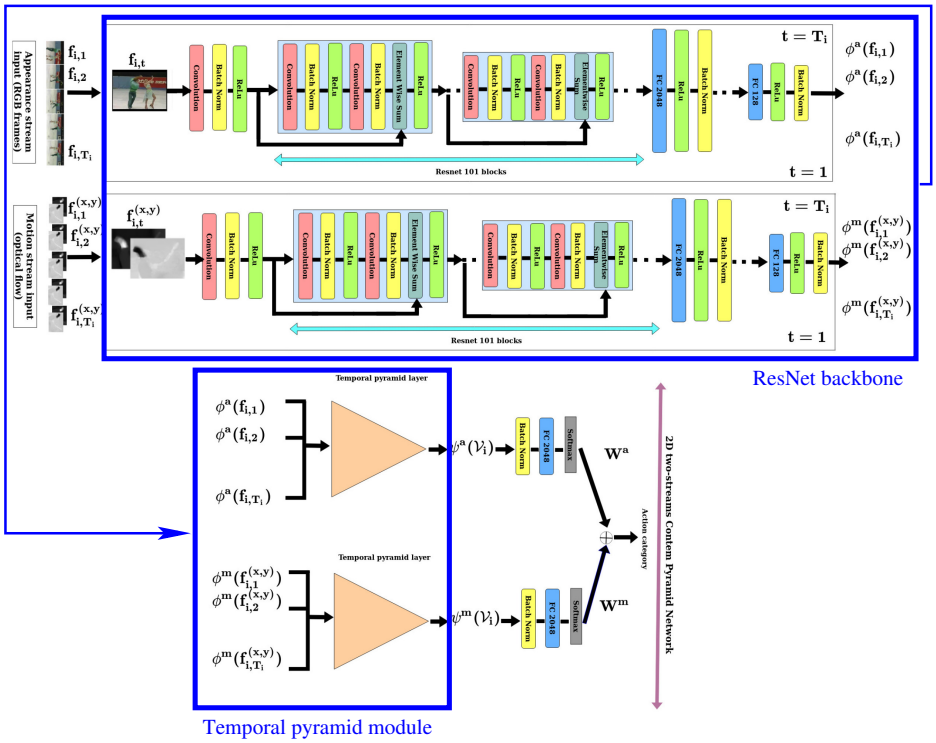


Figure 2: *Our two stream network including a ResNet block, a temporal pyramid block and "batch norm+fully connected+softmax+late fusion" layers. The temporal pyramid block achieves aggregation/pooling as shown in Eq. 3. (***Better to zoom the PDF version***).*

which are linearly transformed so that their range belongs to $[0, 255]$. Following [2, 17], we adapt the inputs of the pretrained ResNet-101 to optical flow data[1]. The number of channels is reset to 20 (instead of 3 in the original ResNet); given a video $\mathcal{V}_i$, these 20 motion channels correspond to stacked optical flow maps extracted from a sliding temporal cube of consecutive frames in $\mathcal{V}_i$ (see input of the motion stream in Fig. 2). The initial weights of these 20 channels are obtained by averaging the original appearance weights and by replicating their values through the 20 new motion channels.

Given a video $\mathcal{V}_i$ (written simply as $\mathcal{V}$) with T frames, we define $\mathcal{N}$ as a tree-structured network with depth up to D levels and width up to 2^{D-1}. Let $\mathcal{N} = \cup_{k,l}\mathcal{N}_{k,l}$ with $\mathcal{N}_{k,l}$ being the k^{th} node of the l^{th} level of $\mathcal{N}$; all nodes belonging to the l^{th} level of $\mathcal{N}$ define a partition of the temporal domain $[0, T]$ into 2^{l-1} equally-sized subdomains (see Fig. 3). A given node $\mathcal{N}_{k,l}$ in this hierarchy aggregates the frames that belong to its underlying temporal interval.

[1] Already available/pretrained on ImageNet to capture the appearance.

Each node $\mathcal{N}_{k,l}$ also defines an appearance and a motion representation respectively denoted as $\psi^a_{k,l}(\mathcal{V}_i)$, $\psi^m_{k,l}(\mathcal{V}_i)$ and set as $\psi^a_{k,l}(\mathcal{V}_i) = \frac{1}{|\mathcal{N}_{k,l}|}\sum_{t\in\mathcal{N}_{k,l}}\phi_a(f_{i,t})$, $\psi^m_{k,l}(\mathcal{V}_i) = \frac{1}{|\mathcal{N}_{k,l}|}\sum_{t\in\mathcal{N}_{k,l}}\phi_m(f_{i,t})$. Depending on the level in $\mathcal{N}$, each representation captures a particular temporal granularity of motion and appearance into a given scene; it is clear that top-level representations capture coarse visual characteristics of actions while bottom-levels (including leaves) are dedicated to fine-grained and timely-resolute sub-actions. Knowing a priori which levels (and nodes in these levels) capture the best – a given action category – is not trivial. In the subsequent section, we introduce a novel learning framework which achieves multiple aggregation design and finds the best combination of levels and nodes in these levels that fits different temporal granularities of action categories.

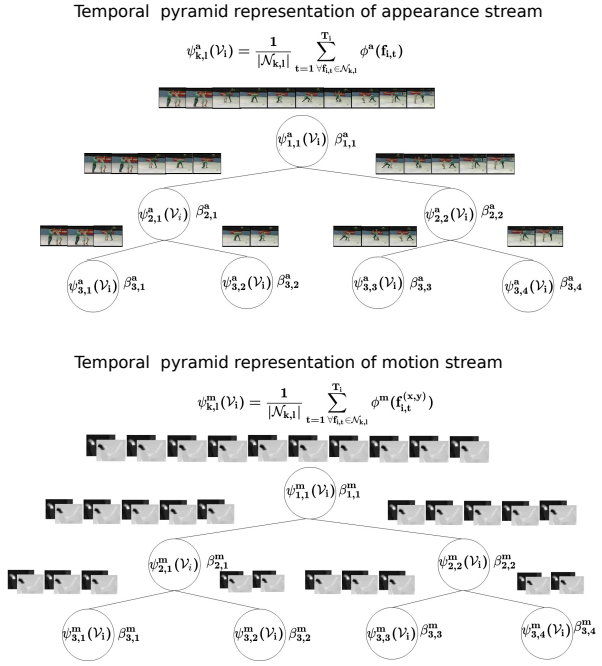


Figure 3: *This figure shows frame aggregation at each node of the temporal pyramid for appearance (top) and motion streams (bottom).*

3 Hierarchical Aggregation Design

In what follows, and unless explicitly mentioned, the symbols m, a are omitted in the notation and all the subsequent formulation is applicable to motion as well as appearance streams.

Given a set of action categories $\mathcal{C} = \{1, \ldots, C\}$, we train multiple classifiers (denoted $\{g_c\}_{c \in \mathcal{C}}$) on top of the level-wise representations $\{\psi_{k,l}(\mathcal{V}_i)\}_{k,l}$. In practice, we use maximum margin classifiers whose kernels correspond to combinations of elementary kernels dedicated to $\{\mathcal{N}_{k,l}\}_{k,l}$. These classifiers are suitable choices as they allow us to weight the impact of nodes in the hierarchy $\mathcal{N}$ and put more emphasis on the most relevant granularity of the learned representations. Hence, depending on the granularity of action categories, these classifiers will prefer top or deep layers of $\mathcal{N}$.
Considering a training set of videos $\{(\mathcal{V}_i, y_{ic})\}_i$ associated to an action category $c \in \mathcal{C}$, with $y_{ic} = +1$ if $\mathcal{V}_i$ belongs to the category c and $y_{ic} = -1$ otherwise, the max margin classifier associated to this action category c is given by $g_c(\mathcal{V}) = \sum_i \alpha_i^c y_{ic} \mathcal{K}(\mathcal{V}, \mathcal{V}_i) + b_c$, here b_c is a shift, $\{\alpha_i^c\}_i$ is a set of positive parameters and $\mathcal{K}$ is a positive semi-definite (p.s.d) kernel [76]. In order to combine different nodes in the hierarchy $\mathcal{N}$ and hence design appropriate aggregation, we consider an extension of multiple kernel learning [18]. Its main idea consists in finding a kernel $\mathcal{K}$ as a combination of p.s.d elementary kernels $\{\kappa(.,.)\}$ associated to $\{\mathcal{N}_{k,l}\}_{k,l}$ as

$$\mathcal{K}(\mathcal{V}, \mathcal{V}') \quad = \quad \sum_l \sum_k \beta_{k,l} \; \kappa(\psi_{k,l}(\mathcal{V}), \psi_{k,l}(\mathcal{V}')), \tag{1}$$

with $\beta_{k,l} \in [0, 1]$ and $\sum_{k,l} \beta_{k,l} = 1$. Here $\beta_{k,l}$ measures the importance (and hence the contribution) of $\psi_{k,l}(\mathcal{V})$ in the global motion representation of $\mathcal{V}$ (denoted as $\psi(\mathcal{V})$). This global representation $\psi(\mathcal{V})$ results from the closure of the p.s.d of κ w.r.t. the sum and the product, so $\mathcal{K}$ can be written as

$$\mathcal{K}(\mathcal{V}, \mathcal{V}') = \big\langle \psi(\mathcal{V}), \psi(\mathcal{V}') \big\rangle, \tag{2}$$

with

$$\psi(\mathcal{V}) \quad = \Big(\sqrt{\beta_{1,1}} \psi_{1,1}(\mathcal{V}) \ldots \sqrt{\beta_{k,l}} \psi_{k,l}(\mathcal{V}) \ldots \Big)^\top. \tag{3}$$

Using the maximum margin formulation, we find the parameters $\beta = \{\beta_{k,l}\}_{k,l}$ and $\{\alpha_i^c\}_{i,c}$ by minimizing the following loss function (denoted as E)

$$\begin{array}{ll} \min\limits_{0 \leq \beta \leq 1, \|\beta\|_1 = 1, \{\alpha_i^c\}} & \dfrac{1}{2} \sum\limits_c \sum\limits_{i,j} \alpha_i^c \alpha_j^c y_{ic} y_{jc} \mathcal{K}(\mathcal{V}_i, \mathcal{V}_j) - \sum\limits_i \alpha_i^c \\ \text{s.t.} & \alpha_i^c \geq 0, \qquad \sum\limits_i y_{ic} \alpha_i^c = 0, \quad \forall i, c. \end{array} \tag{4}$$

As the problem in Eq. 4 is not convex w.r.t β, $\{\alpha_i^c\}$ taken jointly and convex when taken separately, an EM-like iterative optimization procedure could be used: first, one may fix the parameters in β and solve the above problem w.r.t. the classifier parameters $\{\alpha_i^c\}$ using quadratic programming (QP), then one may fix $\{\alpha_i^c\}$ and solve the resulting problem w.r.t. β using linear programming. This iterative process should stop when the values of all these parameters remain unchanged or when it reaches a maximum number of iterations. However, this

EM-like procedure is sub-optimal as it decouples the learning of β from the other parameters. Besides, it requires solving multiple instances of constrained quadratic problems[2] and the number of necessary iterations to reach convergence could be large in practice. We consider instead an end-to-end framework that learns the two-stream parameters β_m and β_a together with classifier and ResNet parameters (denoted as γ_a, γ_m) as well as mixing parameters (referred to as $\mathbf{w}_a$, $\mathbf{w}_m$); the latters capture the importance of appearance and motion streams in action recognition.
Considering E as the loss associated to appearance and motion streams, we find the optimal $\{\beta_m, \beta_a\}$, $\{\gamma_m, \gamma_a\}$ and $\{\mathbf{w}_m, \mathbf{w}_a\}$ by solving

$$\begin{aligned} \min_{\{\beta_s, \gamma_s, \mathbf{w}_s\}_s} \quad & \sum_{s \in \{a,m\}} \mathbf{w}_s \; E(\beta_s, \gamma_s) \\ \text{s.t.} \quad & \|\beta_s\|_1 = 1, \quad 0 \leq \beta_s \leq 1, \quad s \in \{a, m\}. \end{aligned} \tag{5}$$

This objective function can be solved using gradient descent and backpropagation. However, gradient backpropagation (through our multiple aggregation block) should be achieved while considering videos with a varying number of frames. Besides, constraints on $\beta' s$ should also be handled. In what follows, we discuss all these updates in the optimization process.

3.1 Reparametrization trick

Considering $\rho()$ as the final softmax layer of our deep network and considering $\frac{\partial E}{\partial \rho}$ available, the gradient $\frac{\partial E}{\partial \mathbf{w}}$ could easily be obtained by applying the chain rule, in contrast to $\frac{\partial E}{\partial \beta}$, $\frac{\partial E}{\partial \gamma}$. On the one hand, any step following the gradient $\frac{\partial E}{\partial \beta}$ should preserve equality and inequality constraints in Eq. (5) while a direct application of the chain rule provides us with a surrogate gradient which ignores these constraints. On the other hand, the variable number of frames for different training videos requires a careful update of $\frac{\partial E}{\partial \gamma}$ as shown subsequently.
In order to implement equality and inequality constraints when optimizing (5), we consider a re-parametrization as $\beta_{k,l} = h(\hat{\beta}_{k,l}) / \sum_{q=1}^{D} \sum_{p=1}^{2^{q-1}} h(\hat{\beta}_{p,q})$ for some $\{\hat{\beta}_{k,l}\}$ with h being strictly monotonic real-valued positive function and this allows free settings of the parameters $\{\hat{\beta}_{k,l}\}$ during optimization while guaranteeing $\beta_{k,l} \in [0, 1]$ and $\sum_{k,l} \beta_{k,l} = 1$. During back-propagation, the gradient of the loss E (now w.r.t $\hat{\beta}$'s) is updated using the chain rule as

$$\begin{aligned} \frac{\partial E}{\partial \hat{\beta}_{k,l}} \quad &= \quad \sum_{p,q} \frac{\partial E}{\partial \beta_{p,q}} . \frac{\partial \beta_{p,q}}{\partial \hat{\beta}_{k,l}} \\ \text{with} \quad & \frac{\partial \beta_{p,q}}{\partial \hat{\beta}_{k,l}} = \frac{h'(\hat{\beta}_{k,l})}{\sum_{k',l'} h(\hat{\beta}_{k',l'})} . (\delta_{p,q,k,l} - \beta_{p,q}), \end{aligned} \tag{6}$$

[2] whose complexity scales quadratically w.r.t. the size of training data.

and $\delta_{p,q,k,l} = 1_{\{(p,q)=(k,l)\}}$ with $1_{\{.\}}$ being the indicator function. In practice $h(.) = \exp(.)$ and $\frac{\partial E}{\partial \beta_{p,q}}$ is obtained from layerwise gradient backpropagation (as already integrated in standard deep learning tools including PyTorch). Hence, $\frac{\partial E}{\partial \hat{\beta}_{k,l}}$ is obtained by multiplying the original gradient $\left[\frac{\partial E}{\partial \beta_{p,q}}\right]_{p,q}$ by the Jacobian $\left[\frac{\partial \beta_{p,q}}{\partial \hat{\beta}_{p,q}}\right]_{p,q,k,l}$ which merely reduces to $\left[\beta_{k,l}(\delta_{p,q,k,l} - \beta_{p,q})\right]_{p,q,k,l}$.

3.2 Video-length agnostic training

As discussed earlier, motion and appearance ResNets are *recurrently* (iteratively) applied frame-wise prior to pool the underlying representations using multiple aggregation. It is clear that the number of frames intervening in this aggregation is video-dependent, and thereby the number of terms in these aggregations (and the number of ResNet branches/instances) is also varying. Hence, a straightforward application of the chain rule to the whole architecture – in order to update $\frac{\partial E}{\partial \gamma}$ – becomes possible only when this architecture is unfolded, and this requires fixing the maximum number of frames (denoted as T) and sampling temporally all the videos in order to make T_i constant and equal to T. Note that beside requiring all the ResNet instances to share the same parameters (as in Siamese nets), this results into a cumbersome architecture even for reasonable T values. Furthermore, frame sampling requires interpolation techniques which are highly dependent on quality, duration and temporal resolution of videos and this may result into spurious motion/appearance details (especially on short videos; even when timely well resolute) which may ultimately lead to a significant drop in action recognition performances.

In order to avoid these drawbacks and to fully benefit from the available number (and also temporal resolution) of frames — without using multiple instances of "Siamese-like" ResNets and without resampling — we consider an alternative gradient estimation. The latter relies on a membership function μ which assigns each frame to a unique node in the temporal pyramid as $\mu_{i,t}^{k,l} = 1_{\{t \in \mathcal{N}_{k,l}\}}$. Using this membership function, the gradient of the loss E w.r.t. the parameters of the ResNet γ can be updated as

$$\frac{\partial E}{\partial \gamma} = \sum_{k,l} \sum_{i,t} \mu_{i,t}^{k,l} \frac{\partial E}{\partial \psi_{k,l}} \frac{\partial \psi_{k,l}}{\partial \gamma}. \quad (7)$$

From the above equation, it is clear that when $k = l = 1$, all the frames $\{f_{i,t}\}$ contribute in the estimation of the gradient, while for other nodes, only a subsets of frames (belonging to these nodes) are used. However, all the frames contribute equally through all the nodes and hence in gradient estimate, without any sampling. Note also that this formulation implicitly implements *weight sharing* as the above gradient can be rewritten as the sum of gradients, with each one shared across all the frames.

3.3 Efficiency

Taking all the frames during backpropagation, comes at the expense of a substantial increase of computation. This high cost results from the large number of visited frames when (re)estimating the gradient in Eq. 7 w.r.t. the parameters of the ResNet and through epochs of backpropagation. In order to make the evaluation of Eq. 7 (and hence training) more tractable, and with a controlled loss in classification performances, we consider a surrogate gradient as

$$\frac{\partial E}{\partial \gamma} \quad = \quad \sum_{k,l,i} \sum_{t \in \mathcal{P}_r^i} \mu_{i,t}^{k,l} \, \frac{\partial E}{\partial \psi_{k,l}} \, \frac{\partial \psi_{k,l}}{\partial \gamma}, \tag{8}$$

here $\mathcal{P}_r^i$ stands for a subset of selected frames, in a given video $\mathcal{V}_i$, that contribute to gradient estimation at the r^{th} epoch. We consider a periodic selection mechanism which guarantees that all the frames are equally used through epochs; in practice, $\mathcal{P}_r^i = \{t \in [0, T_i],\ t \equiv r \pmod K\}$ with $1/K$ being the fraction of frames used per epoch. With this mechanism, gradient evaluation still relies on the entire set of frames in the training set, but their use is distributed through epochs and this makes the evaluation and training process far more efficient while maintaining close performances (see experiments).

4 Experiments

We evaluate the performance of our temporal pyramid design on three standard action recognition datasets: UCF-101, HMDB-51 and JHMDB-21 [19, 20]. The largest and most challenging one (UCF-101) is used to comprehensively study different settings of our model. It includes 13320 videos belonging to 101 categories with variable duration, poor frame resolution, viewpoint and illumination changes, occlusion, cluttered background and eclectic content ranging from multiple and highly interacting individuals to single and completely passive ones. We also consider HMDB-51 and JHMDB-21 for further comparisons; the latter include 6766 (resp. 928) videos belonging to 51 (resp. 21) action categories. In all these experiments, we process all the videos using ResNet-101 in order to extract appearance and motion representations framewise. Then, we apply different aggregation schemes prior to assign those videos to classes. We use the same evaluation protocols as the ones suggested in [19, 20] (i.e., train/test splits) and we report the average accuracy over all the categories of actions.

We train our temporal pyramid-based networks for respectively 130, 100 and 65 iterations on UCF-101, HMDB-51 and JHMDB-21 using the PyTorch SGD optimizer. For appearance stream, we set the learning rate to 0.001 and reduce it by a factor of 10 every 25, 20, 10 iterations for resp. UCF-101, HMDB-51 and JHMDB-21. For motion stream, we set the learning rate to 0.005 and we reduce it by the same factor after "80 and 110", "60 and 80", "50 and 60" iterations on the three sets respectively. Experiments on individual streams are run using 4 Titan X Pascal GPUs (with 12 GB) and last 72h for UCF101, 36h for HMDB-51 and 15h for JHMDB-21 on the appearance stream. On the motion stream, these

experiments last 96h for UCF101, 48h for HMDB-51 and 24h for JHMDB-21 while on the joint (motion+appearance) stream experiments are run using 4 Tesla P100 GPUs (with 16 GB) and last 100h, 55h and 30h on the three sets respectively.

4.1 Model analysis

Impact of pyramid depth and fusion. Experiments, reported in Fig. 4-left, show that our temporal pyramid selects the best configurations (combinations) of level representations that improve the performance of action recognition; indeed, the results show a clear gain as the depth of the pyramid increases. This gain results from the match between the temporal granularity of the learned level-wise representations and the actual granularity of action categories. In all these results, multi-level combinations (levels 2 up to 6) provide a clear and a consistent gain w.r.t. global averaging (level 1) both on motion and appearance streams as well as their fusion. As already discussed, the parameters $\mathbf{w}_a$, $\mathbf{w}_m$ of this fusion are optimized as a part of the end-to-end learning process. Results reported in Fig. 4-right show these two parameters $\mathbf{w}_a$, $\mathbf{w}_m$ w.r.t. the depth of the temporal pyramid and hence the complementary aspects of the two streams. We observe that the contribution of the motion stream is strictly increasing as the level of the temporal pyramid increases (and a contrario strictly decreasing for appearance stream). This clearly corroborates the highest impact of motion in the combined setting when modeling the temporal granularity of action categories (see also Fig. 5).

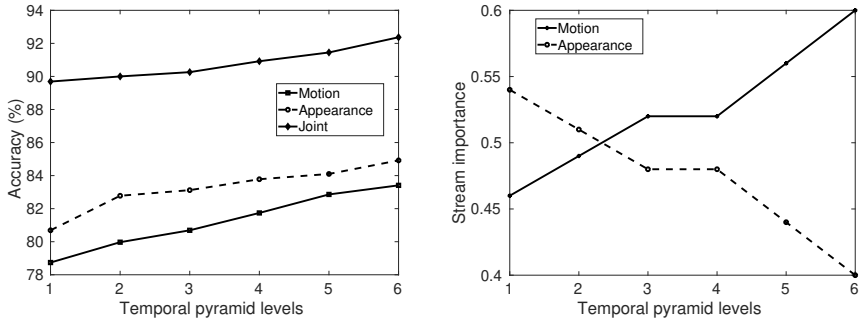


Figure 4: *This figure shows level-wise performances of motion, appearance and joint streams (left) and also the importance of each stream (i.e., obtained* $\mathbf{w}_a$ *and* $\mathbf{w}_m$ *values) when fusing motion and appearance (right).*

Impact of multiple pyramids. We further investigate the potential of our method using multiple instances of temporal pyramids jointly trained (see Table 1). Performances reported in this table show a small gain when multiple pyramids are combined (concatenated) and this results from the heterogeneity of action categories and their dynamics which may require multiple pooling mechanisms (i.e., different pyramids). Indeed, the apex of some actions appears

# of temporal pyramids per stream	Accuracy		
	Appearance stream	Motion stream	Joint stream
1	83.92	81.69	90.78
2	83.95	81.73	90.79
4	**83.97**	81.79	90.84
8	83.92	**81.86**	**90.89**
16	83.89	81.83	90.85

Table 1: *This table shows the evolution of the performances w.r.t. different # of temporal pyramids per stream. In order to combine the outputs of these multiple pyramids (when using concatenation), we add a succession of FC+Relu+BatchNorm to reduce the dimensionality from "63 (number of nodes in TP of 6 levels) × 128 (node dimension) × # TPs" to "128". The choice of FC layer of 128 dimensions (for multiple pyramids) is made in order to reduce time and memory footprint while maintaining relatively high accuracy. All these results correspond to temporal pyramids of 6 levels.*

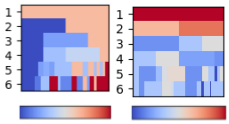
early in video clips while for others later or spread through all the video duration. Hence, instead of learning a single monolithic temporal pyramid per stream, one may stack multiple instances of temporal pyramids (with different weights β) with each one dedicated to a subclass of actions whose dynamics are similar[3]. However, as observed in Table 1, the gain is marginal and this is explained by the correlation of the learned weights for different pyramids as shown in Fig. 5 (b) and (c).

Sampling settings	# frames (train)		# frames (test)		Accuracy		
	RGB	OF	RGB	OF	Appearance	Motion	Fusion
#1	25	25	25	25	84.23	81.27	91.65
#2	25	25	25	250	84.23	81.27	91.64
#3	25	50	25	50	84.23	81.86	91.69
#4	25	50	25	250	84.23	81.89	91.78
#5	64	64	250	250	84.62	82.05	91.89
#6	64	64	all	all	84.81	82.77	92.09
#7	64	all	all	all	84.81	83.41	92.29
#8	all	all	all	all	84.92	83.41	92.37

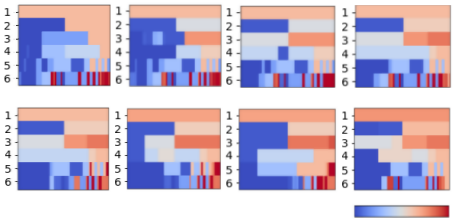
Table 2: *This table shows the evolution of the performance w.r.t. to different sampling settings (i.e., number of frames in training and test videos). RGB and OF stand for the number of input RGB frames and the number of optical flow frames used in the appearance and the motion streams respectively. These performances are obtained using a temporal pyramid of six levels.*

Sampling and efficiency. Table. 2 shows the impact of our method – with and without frame sampling – on the performance of action recognition. These

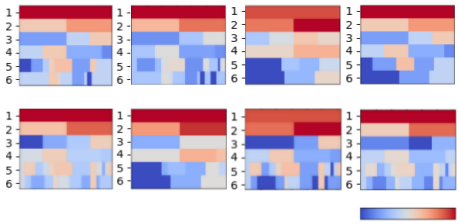
[3]These subclasses of actions are not explicitly defined in a supervised manner but implicitly by allowing enough flexibility in the multiple instances of temporal pyramids in order to capture different (unknown) subclasses of action dynamics.



(a) Single temporal pyramid



(b) Multiple temporal pyramids (motion stream)



(c) Multiple temporal pyramids (appearance stream)

Figure 5: *(a) Weight distribution of motion and appearance streams obtained when learning the parameters of a single temporal pyramid (corresponding to the first row of table 1). (b-c) Weight distribution of multiple temporal pyramids of motion and appearance streams (corresponding to the fourth row in the same table); the y-axes correspond to pyramid levels (from 1 up to 6) while the x-axes correspond to nodes in these levels. Warmer colors correspond to higher weights while cooler colors to lower ones.*

Training time motion, appear, comb	Speed up factor (K)	Avg. # of frames (train)	Accuracy Appearance	Motion	Joint
96h, 72h, 100h (4 days)	1×	185	**84.92**	**83.41**	**92.37**
24h, 18h, 25h (1 day)	4×	92	84.27	82.59	91.74
12h, 9h, 12h (half-day)	8×	46	84.10	82.07	91.39
6h, 4h, 6h	16×	23	83.96	81.23	90.70
4h, 3h, 4h	24×	8	83.89	80.95	90.35

Table 3: *This table shows the performance of "surrogate back-propagation" with different acceleration factors. Note that motion stream performances are more sensitive to this acceleration compared to appearance stream.*

results are obtained using a single pyramid. From these results, it is easy to see that performances get better as the number of sampled frames increases reaching asymptotically the best performances when all the frames are used. This behavior is similar both on motion and appearance streams. However, we notice that motion stream, which is based on optical flow data, is more sensitive to sampling than appearance stream so the accuracy of the former is clearly proportional to the number of frames. In other words, motion stream builds a better representation and hence becomes more important for the overall action classification when it is fed with more optical flow data as shown again in Table 2 (settings #6 and #7). Nonetheless, taking all the frames during backpropagation, comes at the expense of a substantial increase of computation; for instance when considering all the 2.5 millions frames of our videos on UCF-101, training costs 72h (resp. 96h) for appearance (resp. motion) stream using 4 Titan X GPUs (with 12 GB) and 100h on the joint stream using 4 Tesla P100 GPUs (with 16 GB); see Table. 3. This high cost results from the large number of visited frames when (re)estimating the gradient in (7). Eq. 8 reduces substantially this cost by considering fractions $\frac{1}{K}$ of frames during backpropagation; for instance, when $K = 24$, training is $24\times$ faster compared to the most accurate setting (setting #8 in Table 2) as only 8 frames are used (on average "per epoch-per video") in Eq. 8 instead of 185. Moreover, as all frames contribute equally through all the epochs, the loss in accuracy is contained. These performances are obtained on individual and joint streams using the same aforementioned hardware resources.

4.2 Ablation study and comparison

Table. 4 shows an ablation study of our complete model when "network retraining" and "temporal pyramid" are taken individually and combined. As observed from the motion stream, retraining the baseline ResNet improves the performance by +0.34 while temporal pyramid improves the performance by +1.56. Combining both "retraining" and "temporal pyramid" brings a substantial gain of +5.01. Similarly to motion, the gain with "retraining" and "temporal pyramid" when taken individually and combined reaches +0.41, +1.68 and +4.64 respectively and a similar trend is observed when combining both motion and appearance streams. In sum, the gain brought by our temporal pyramid is clearly established especially when the ResNet backbone is retrained (fine-tuned).

We also compare the performance and the complementary aspects of our temporal pyramid model w.r.t. the related state-of-the art methods [16, 71, 29, 7] on UCF-101, HMDB-51 and JHMDB-21. The comparisons are shown for different 2D and 3D convolutional backbones as well as pooling strategies. All these networks are based on 2D and 3D spatio-temporal filters [71, 7] that consider motion and appearance streams and their design is end-to-end but clearly differ in their pooling mechanisms and the way frames are exploited. Indeed, these related techniques rely on sampling strategies that vectorize video sequences into fixed length inputs while our method keeps all the frames in order to build temporal pyramids. Another major difference w.r.t. our method resides in the huge set used in order to train these related architectures. Pooling

Configurations	Accuracy
Motion stream	
ResNet-101 baseline	78.40
ResNet-101 + full network retraining	78.74
ResNet-101 + temporal pyramid	79.96
ResNet-101 + full network retraining + temporal pyramid	**83.41**
Appearance stream	
ResNet-101 baseline	80.28
ResNet-101 + full network retraining	80.69
ResNet-101 + temporal pyramid	81.96
ResNet-101 + full network retraining + temporal pyramid	**84.92**
Joint (motion and appearance) stream	
ResNet-101 baseline	88.91
ResNet-101 + full network retraining	89.69
ResNet-101 + temporal pyramid	89.26
ResNet-101 + full network retraining + temporal pyramid	**92.37**

Table 4: This table shows an ablation study of our model involving "network retraining" and "temporal pyramid" settings.

mechanisms used on top of these backbones include global averaging and spectrograms. The former produces a global representation that averages all the frame representations while the latter keeps all the frame representations and concatenate them prior to their classifications (see Fig. 6). Note that these two settings are related to the two extreme cases of our hierarchy, i.e., the root and the leaves. In particular, the spectrogram of a video $\mathcal{V}$ with T frames is obtained when the number of leaf nodes, in the hierarchy, is exactly equal to T. Global averaging techniques (shown in Table. 5) also include [16]; the latter, based on colorized heatmaps, correspond to timely-stamped and averaged framewise probability distributions of human keypoints. These colorized heatmaps are fed to a 2D CNN for classification; note that colorized heatmaps provide video-level representations which capture globally the dynamics of video actions without any weighting scheme to emphasize the most important temporal granularities of these actions and this results into low accuracy as again displayed in Table. 5. While these 2D backbones are already effective when combined with global average pooling and/or spectrograms, their combination with our temporal pyramid brings an extra significant gain. We observe the same behavior on the 3D backbones. Note that some of these backbones rely on extra datasets (including Kinetics) in order to pretrain their CNNs while our method is trained only on the original datasets of UCF-101, HMDB-51 and JHMDB-21.

4.3 Extra comparison

In this section, we also evaluate the performances of our proposed temporal pyramid on the task of skeleton-based action recognition using a challenging dataset, namely FPHA [91]. The latter includes 1175 videos (with 3D skeleton and RGB frames as well as depth information[4]) belonging to 45 action cate-

[4]In order to make training cycles efficient, we only use the skeleton frames.

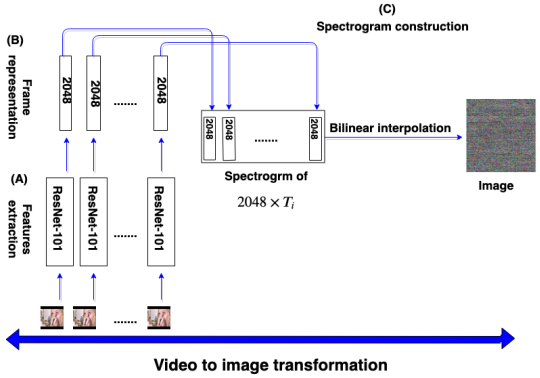


Figure 6: *This figure shows the general scheme of "spectrogram" construction.*

gories with high inter and intra subject variability. Each video corresponds to a sequence of skeleton frames, each one includes 21 hand joints, and each joint is encoded with its 3D coordinates. We evaluate the performance of our method following the protocol in [91]. In all these experiments, we report the average accuracy over all the classes of actions.

In order to achieve action recognition, we use a baseline graph convolutional network (GCN) architecture as a backbone (similar to [54, 68, 87]). This architecture includes an attention layer of 16 heads applied to skeleton graphs whose nodes are encoded with 3-channels (3D joint coordinates), followed by a convolutional layer of 32 filters, and a dense fully connected layer. This baseline GCN network is relatively lightweight, its number of parameters does not exceed 239976, and this makes its training and testing cycles highly efficient while being accurate[5]. We train all our GCNs — with different pooling mechanisms, namely temporal pyramid (TP), global average pooling (GAP) and spectrogram (Spect) — end-to-end using the Adam optimizer [92] for 2,700 epochs with a batch size equal to 600, a momentum of 0.9 and a global learning rate (denoted as $\nu(t)$) inversely proportional to the speed of change of the classification loss used to train our networks. When this speed increases (resp. decreases), $\nu(t)$ decreases as $\nu(t) \leftarrow \nu(t-1) \times 0.99$ (resp. increases as $\nu(t) \leftarrow \nu(t-1)/0.99$). As shown in Table. 6, when combining this GCN with our TP pooling, it outperforms both GCNs with GAP and Spect. It also shows very competitive results compared to the closely related *hierarchical attention network (HAN)* work in [90] while being very lightweight and accurate. Indeed, when combining our GCN with TP, the number of parameters of our lightweight GCN increases from 239976 (without TP) to 240648 only (with TP), which is four times smaller than HAN-2S and twice smaller than HAN [90][6] while also being equivalently competitive.

[5]Training of each lightweight GCN architecture lasts less than an hour on a GeForce GTX 1070 GPU (with 8 GB memory).

[6]As reported in [90], the number of parameters in HAN-2S and HAN architectures are 940k and 530k respectively.

Methods	UCF-101	HMDB-51	JHMDB-21	ImageNet pretraining	Kinetics pretraining
ResNet-A + Spect [29]	78.40	57.76	61.26	Yes	No
ResNet-M + Spect [29]	76.46	55.38	60.66	Yes	No
ResNet-J + Spect [29]	80.10	58.28	62.14	Yes	No
ResNet-M + GAP [71]	79.4	59.13	61.39	Yes	No
ResNet-A + GAP [71]	82.1	60.24	62.71	Yes	No
ResNet-J + GAP [71]	88.5	63.31	64.11	No	No
ResNet-A + Our TP	**83.41**	**61.04**	**62.97**	Yes	No
ResNet-M + Our TP	**84.92**	**62.23**	**63.51**	Yes	No
ResNet-J + Our TP	**92.37**	**65.14**	**66.96**	Yes	No
2D col-heat [16]	64.38	54.90	60.5	No	No
2D col-heat + Our TP	**80.41**	**65.21**	**69.93**	No	No
3D CNN-M [7]	96.41	80.39	-	Yes	Yes
3D CNN-A [7]	**95.60**	76.47	-	Yes	Yes
3D CNN-M + Our TP	**96.61**	**80.54**	-	No	No
3D CNN-A + Our TP	96.05	**76.56**	-	No	No
Other Action Recognition Methods					
TwoStr (2014) [2]	88.0	59.4	-	Yes	No
TwoStrF (2014) [2]	92.5	65.4	-	Yes	No
C3Dr (2015) [69]	82.3	51.6	-	-	-
ST-Res (2016) [14]	93.4	66.4	-	Yes	No
TSNr (2016) [17]	94.0	68.5	-	Yes	No
P3Dr (2018)[104]	88.6	-	-	-	-
Res3Dr (2018) [105]	85.8	54.9	-	-	-
R(2 + 1)Dr (2018) [105]	93.6	66.6	-	-	-
ARTNetr (2018) [106]	94.3	70.9	-	No	Yes
ECOr (2018) [107]	94.8	72.4	-	No	Yes
CoViARr (2018) [102]	94.9	70.2	-	Yes	No
I3Dr (2017) [7]	95.6	74.8	-	Yes	Yes
TSMr (2019) [109]	95.9	73.5	-	Yes	Yes
NST (2021) [110]	96.0	76.1	-	Yes	Yes

Table 5: *This table shows a comparison of our temporal pyramid (referred to as TP) w.r.t. different related works; in this table, "col-heat" stands for colorized heatmaps, "Spect" for spectrograms, "A" for appearance, "M" for motion and "GAP" for global average pooling. In our experiments, ResNets are pretrained on ImageNet and fine-tuned on UCF-101 (for both appearance and motion).*

5 Conclusion

We introduce in this paper a temporal pyramid approach for video action recognition. The strength of the proposed method resides in its ability to learn hierarchical pooling operations that capture different levels of temporal granularity in action recognition. This is translated into learning the distribution of weights in the temporal pyramid that capture these granularities. This is obtained by solving a constrained quadratic programming problem, and by optimizing the parameters of a deep network including a temporal pyramid module both on motion and appearance streams as well as their combination. We also consider variants of the deep learning framework that designs multiple instances

Method	Color/RGB	Depth	Skeleton	Accuracy (%)
Two stream-color (2016) [6]	✓	✗	✗	61.56
Two stream-flow (2016) [6]	✓	✗	✗	69.91
Two stream-all (2016) [6]	✓	✗	✗	75.30
HOG2-depth (2014) [77]	✗	✓	✗	59.83
HOG2-depth+pose (2014) [77]	✗	✓	✓	66.78
HON4D (2013) [78]	✗	✓	✗	70.61
Novel View (2016) [80]	✗	✓	✗	69.21
1-layer LSTM (2016) [81]	✗	✗	✓	78.73
2-layer LSTM (2016) [81]	✗	✗	✓	80.14
Moving Pose (2013) [82]	✗	✗	✓	56.34
Lie Group (2014) [83]	✗	✗	✓	82.69
HBRNN (2015) [84]	✗	✗	✓	77.40
Gram Matrix (2016) [85]	✗	✗	✓	85.39
TF (2017) [86]	✗	✗	✓	80.69
JOULE-color (2015) [88]	✓	✗	✗	66.78
JOULE-depth (2015) [88]	✗	✓	✗	60.17
JOULE-pose (2015) [88]	✗	✗	✓	74.60
JOULE-all (2015) [88]	✓	✓	✓	78.78
Huang et al. (2017) [89]	✗	✗	✓	84.35
Huang et al. (2018) [93]	✗	✗	✓	77.57
HAN (2021) [90]	✗	✗	✓	85.74
HAN-2S (2021) [90]	✗	✗	✓	89.04
GCN + Spect	✗	✗	✓	86.78
GCN + GAP	✗	✗	✓	87.13
GCN + Our TP	✗	✗	✓	87.47

Table 6: Comparison of our GCNs, with different poolings, against related work on FPHA.

of temporal pyramids each one dedicated to a particular subcategory of action granularities and also a procedure that allows us to efficiently train the network at the detriment of a slight decrease of its classification accuracy. The advantages of these contributions are established, against different baselines as well as the related work, through extensive experiments on challenging action recognition benchmarks including the UCF-101, HMDB-51 and JHMDB-21 datasets as well as skeleton videos including the FPHA dataset.

References

[1] Martin, Pierre-Etienne, Jenny Benois-Pineau, Renaud Péteri, and Julien Morlier. "Fine grained sport action recognition with Twin spatio-temporal convolutional neural networks: Application to table tennis." Multimedia Tools and Applications 79 (2020): 20429-20447.

[2] K. Simonyan, A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In Neural Information Processing Systems (NeurIPS), 2014

[3] Obeso, Abraham Montoya, Jenny Benois-Pineau, Mireya Saraí García Vázquez, and Alejandro Álvaro Ramírez Acosta. "Forward-backward visual saliency propagation in deep nns vs internal attentional mechanisms." In 2019 Ninth international conference on image processing theory, tools and applications (IPTA), pp. 1-6. IEEE, 2019.

[4] Obeso, Abraham Montoya, Jenny Benois-Pineau, Mireya Saraí García Vázquez, and Alejandro A. Ramírez Acosta. "Introduction of explicit visual saliency in training of deep cnns: Application to architectural styles classification." In 2018 International Conference on Content-Based Multimedia Indexing (CBMI), pp. 1-5. IEEE, 2018.

[5] Pramono, Rizard Renanda Adhi, Yie-Tarng Chen, and Wen-Hsien Fang. "Hierarchical self-attention network for action localization in videos." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 61-70. 2019.

[6] C. Feichtenhofer, A. Pinz, A. Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[7] J. Carreira, A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[8] H. Sahbi and H. Zhan. "FFNB: Forgetting-Free Neural Blocks for Deep Continual Learning." In The British Machine Vision Conference (BMVC). 2021.

[9] H. Pirsiavash, D. Ramanan. Detecting Activities of Daily Living in First-person Camera Views. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012

[10] L. Chen, L. Duan, D. Xu. Event Recognition in Videos by Learning From Heterogeneous Web Sources. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2013

[11] D. Xu, S-F. Chang. Visual Event Recognition in News Video using Kernel Methods with Multi-Level Temporal Alignment. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2007

[12] H. Wang, C. Yuan, W. Hu, C. Sun. Supervised class-specific dictionary learning for sparse modeling in action recognition. Pattern Recognition, Volume 45, Issue 11, Pages 3902-3911, 2012

[13] C. Schuldt, I. Laptev, B. Caputo. Recognizing human actions: a local SVM approach. In IEEE International Conference on Pattern Recognition (ICPR), 2004

[14] C. Feichtenhofer, A. Pinz, R-P. Wildes. Spatiotemporal Residual Networks for Video Action Recognition. In Neural Information Processing Systems (NeurIPS), 2016

[15] C. Feichtenhofer, A. Pinz, R-P. Wildes. Spatiotemporal Multiplier Networks for Video Action Recognition. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2017

[16] V. Choutas, P. Weinzaepfel, J. Revaud, C. Schmid. PoTion: Pose MoTion Representation for Action Recognition. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2018

[17] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In European Conference on Computer Vision (ECCV), 2016

[18] M. Gönen, E. Alpaydın. Multiple Kernel Learning Algorithms. In Journal of Machine Learning Research (JMLR) : 2211-2268, 2011

[19] Amir Roshan Zamir Khurram Soomro and Mubarak Shah, "Ucf101: A dataset of 101 human action classes from videos in the wild," in *CRCV-TR-12-01*, 2012.

[20] 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.

[21] C.cortes, M. Mohri, A. Rostamizadeh. Algorithms for learning Kernels based on Centered Alignement. In Journal of Machine Learning Research (JMLR) : 795-828, 2012

[22] B. Zoph, V. Vasudevan, J. Shlens, Q-V. Le. Learning Transferable Architectures for Scalable Image Recognition. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2018

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2009.

[24] Y. Wang, M. Long, J. Wang, Philip S. Yu. Spatio temporal Pyramid Network for Video Action Recognition. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2017

[25] J. Zhu, W. Zou, Z. Zhu. End-to-end Video level Representation Learning for Action Recognition. In International Conference on Learning Representation (ICLR), 2018

[26] Z. Zheng, G. An, D. Wu, Q. Ruan. Spatial-temporal pyramid based Convolutional Neural Network for action recognition. Neurocomputing, Volume 358, 17 September 2019, Pages 446-455

[27] Zhang D., Dai X., Wang YF. Dynamic Temporal Pyramid Network: A Closer Look at Multi-scale Modeling for Activity Detection. In Asian Conference on Computer Vision (ACCV), 2018

[28] K. Yang, R. Li, P. Qiao, Q. Wang, D. Li, Y. Dou. Temporal Pyramid Relation Network For Video-based Gesture Recognition. In IEEE International Conference on Image Processing (ICIP), 2018

[29] A. Mazari, H. Sahbi. Deep Temporal Pyramid Design for Action Recognition. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019

[30] H. Kaiming, Z. Xiangyu, R. Shaoqing; S. Jian. Deep Residual Learning for Image Recognition. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2016

[31] B. K.P.Horn, B. G.Schunck. Determining optical flow. Artificial Intelligence, Volume 17, Issues 1–3, Pages 185-203, 1981

[32] W. Lu and James J. Little. Simultaneous tracking and action recognition using the pca-hog descriptor. In European conference on Computer vision (ECCV), 2006

[33] I. Laptev. On Space-Time Interest Points. In International Journal of Computer Vision (IJCV), Volume 64, Issue 2–3, pp 107–123, 2005

[34] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, S. Savarese. Social Scene Understanding: End-To-End Multi-Person Action Localization and Collective Activity Recognition. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2017

[35] J. Shao, K. Kang, C. Change Loy, X. Wang. Deeply Learned Attributes for Crowded Scene Understanding. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2015

[36] M. Pantic, A. Pentland, A. Nijholt, T.S. Huang. Human Computing and Machine Understanding of Human Behavior: A Survey. In Human Computing and Machine Understanding of Human Behavior, 2007

[37] A. Ben Mabrouk, E. Zagrouba. Abnormal behavior recognition for intelligent video surveillance systems: A review. In Expert Systems with Applications Volume 91, Pages 480-491, 2018

[38] Y. Han, P. Zhanga, T. Zhuob, W. Huang, Y. Zhanga. Going deeper with two-stream ConvNets for action recognition in video surveillance. In Pattern Recognition Letters Volume 107, Pages 83-90, 2018

[39] B. Wang, L. Ma, W. Zhang, W. Liu. Reconstruction network for video captioning. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2018

[40] L. Wang and H. Sahbi. "Nonlinear cross-view sample enrichment for action recognition." In Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III 13, pp. 47-62. Springer International Publishing, 2015.

[41] J. Wang, W. Jiang, L. Ma, W. Liu, Y. Xu. Bidirectional attentive fusion with context gating for dense video captioning. In IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), 2018

[42] N. Aafaq, N. Akhtar, W. Liu, S. Zulqarnain Gilani, A. Mian. Spatio-Temporal Dynamics and Semantic Attribute Enriched Visual Encoding for Video Captioning. In the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019

[43] Minlong Lu, Ze-Nian Li, Yueming Wang, Gang Pan. Deep Attention Network for Egocentric Action Recognition. In IEEE Transactions on Image Processing, Volume 28, Issue 8, 2019

[44] T. Mahmud, M. Billah, M. Hasan, Am. K. Roy-Chowdhury. Captioning Near-Future Activity Sequences. In arXiv:1908.00943, 2019

[45] I. Laptev, P. Perez. Retrieving actions in movies. In International Conference on Computer Vision (ICCV), 2007

[46] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, G. Serra. Event detection and recognition for semantic annotation of video. In Multimedia Tools and Applications, Volume 51, Issue 1, pp 279–302, 2011

[47] A. Jaimes, K. Omura, T. Nagamine, K. Hirata. Memory Cues for Meeting Video Retrieval. In CARPE Proceedings of the 1st ACM workshop on Continuous archival and retrieval of personal experiences, Pages 74-85, 2004

[48] O. Duchenne, I. Laptev, J. Sivic, F. Bach, J. Ponce. Automatic Annotation of Human Actions in Video. In International Conference on Computational Vision (ICCV), 2009

[49] H. Meng, N. Pears, C. Bailey. A Human Action Recognition System for Embedded Computer Vision Application. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007

[50] T. Theodoridis, A. Agapitos, H. Hu, S.M. Lucas. Ubiquitous robotics in physical human action recognition: A comparison between dynamic ANNs and GP. In IEEE International Conference on Robotics and Automation, 2008

[51] Y. Demiris. Prediction of intent in robotics and multi-agent systems. Cogn Process (2007) 8: 151. https://doi.org/10.1007/s10339-007-0168-9

[52] M. Nan, A. Stefania Ghiță, A. Gavril, M. Trascau, A. Sorici, B. Cramariuc, A. Magda Florea. Human Action Recognition for Social Robots. In nternational Conference on Control Systems and Computer Science, 2019

[53] E. Coupeté, F. Moutarde, S. Manitsaris. Multi-users online recognition of technical gestures for natural human–robot collaboration in manufacturing. Robot (2019) 43: 1309. https://doi.org/10.1007/s10514-018-9704-y

[54] H. Sahbi. "Learning connectivity with graph convolutional networks." In 2020 25th International Conference on Pattern Recognition (ICPR), pp. 9996-10003. IEEE, 2021.

[55] K. He, X. Zhang, S. Ren, J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In IEEE International Conference on Computer Vision (ICCV), 2015

[56] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, Z. Zhang. The Application of Two-Level Attention Models in Deep Convolutional Neural Network for Fine-Grained Image Classification. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015

[57] A. Graves, A. Mohamed, G. Hinton. Speech recognition with deep recurrent neural networks. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013

[58] G.y Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath and B. Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition. In IEEE Signal Processing Magazine, Vol 29: pp. 82-97, 2012

[59] S. S. Beauchemin, J. L. Barron. The computation of optical flow. ACM Computing Surveys (CSUR) Surveys, Volume 27, Issue 3, Pages 433-466, 1995

[60] D. Gu, Z. Wen, W. Cui, R. Wang, F. Jiang, S. Liu. Continuous Bidirectional Optical Flow for Video Frame Sequence Interpolation. In IEEE International Conference on Multimedia and Expo (ICME), 2019

[61] D. Sun, S. Roth, M. J. Black. Secrets of optical flow estimation and their principles. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010

[62] L. Xu, J. Jia, Y. Matsushita. Motion Detail Preserving Optical Flow Estimation. In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Volume : 34, Issue : 9, 2012

[63] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray. Visual Categorization with Bags of Keypoints. In European Conference on Computer Vision (ECCV), 2004

[64] G. Csurka, F. Perronnin. Fisher Vectors : Beyond Bag-of-Visual-Words Image Representations. In International Conference on Computer Vision, Imaging and Computer Graphics, 2010

[65] K. He, X. Zhang, S. Ren, J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Volume : 37, Issue : 9 , 2015

[66] N. Murray, F. Perronnin. Generalized Max Pooling. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014

[67] Y. Gao, O. Beijbom, N. Zhang, T. Darrell. Compact Bilinear Pooling. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[68] H. Sahbi. "Lightweight Connectivity In Graph Convolutional Networks For Skeleton-Based Recognition." In IEEE International Conference on Image Processing (ICIP), pp. 2329-2333. 2021.

[69] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In IEEE International Conference on Computer Vision (ICCV), 2015

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

[71] J. Yihuang. Pretrained 2D two streams network for action recognition on UCF-101 based on temporal segment network. https://github.com/jeffreyyihuang/two-stream-action-recognition , 2017

[72] A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In Neural Information Processing Systems (NeurIPS), 2012

[73] AJ Piergiovanni, M.S. Ryoo. Fine-grained Activity Recognition in Baseball Videos. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Computer Vision in Sports, 2018

[74] A. Shahroudy, J. Liu, T. Ng, G. Wang. NTU RGB+D : A Large Scale Dataset for 3D Human Activity Analysis. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[75] Ullah, Amin, et al. "Action recognition in video sequences using deep bi-directional LSTM with CNN features." IEEE Access 6 (2017): 1155-1166.

[76] Shawe-Taylor, John, and Nello Cristianini. Kernel methods for pattern analysis. Cambridge university press, 2004.

[77] E.Ohn-Barand, M.M.Trivedi. Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision- Based Approach and Evaluations. IEEE TITS, 15(6):2368–2377, 2014.

[78] O. Oreifej and Z. Liu. HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In CVPR, pages 716-723, June 2013.

[79] L. Wang and H. Sahbi. "Bags-of-daglets for action recognition." In 2014 IEEE International Conference on Image Processing (ICIP), pp. 1550-1554. IEEE, 2014.

[80] H. Rahmani and A. Mian. 3D Action Recognition from Novel Viewpoints. In CVPR, pages 1506–1515, June 2016.

[81] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks In AAAI, volume 2, page 6, 2016.

[82] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection. In ICCV, pages 2752–2759, 2013.

[83] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a Lie group. In IEEE CVPR, pages 588–595, 2014

[84] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In IEEE CVPR, pages 1110–1118, 2015.

[85] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps. Efficient Temporal Sequence Comparison and Classification Using Gram Matrix Embeddings on a Riemannian Manifold. In CVPR, pages 4498–4507, 2016

[86] G. Garcia-Hernando and T.-K. Kim. Transition Forests: Learning Discriminative Temporal Transitions for Action Recognition. In CVPR, pages 407–415, 2017.

[87] H. Sahbi. "Topologically-Consistent Magnitude Pruning for Very Lightweight Graph Convolutional Networks." In IEEE International Conference on Image Processing (ICIP), pp. 3495-3499. 2022.

[88] J. Hu, W. Zheng, J. Lai, and J. Zhang. Jointly Learning Heterogeneous Features for RGB-D Activity Recognition. In CVPR 2015

[89] Z. Huang and L. V. Gool. A Riemannian Network for SPD Matrix Learning. In AAAI, pages 2036–2042, 2017

[90] Liu, Jianbo, Ying Wang, Shiming Xiang, and Chunhong Pan. "Han: An efficient hierarchical self-attention network for skeleton-based gesture recognition." arXiv preprint arXiv:2106.13391 (2021).

[91] G. Garcia-Hernando, S. Yuan, S. Baek, and T.K. Kim. First Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In CVPR, 2018.

[92] D.P. Kingma, and J. Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014)

[93] Z. Huang, J. Wu, and L. V. Gool. Building Deep Networks on Grassmann Manifolds. In AAAI, pages 3279-3286, 2018

[94] Li, Huifang, Jingwei Huang, Mengchu Zhou, Qisong Shi, and Qing Fei. "Self-attention pooling-based long-term temporal network for action recognition." IEEE Transactions on Cognitive and Developmental Systems (2022).

[95] Jin, Sheng, Zhongqing Cao, and Xingguo Song. "IA-FPN: Interactive Aggregation Feature Pyramid Network for Action Detection." In 2022 4th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), pp. 1063-1068. IEEE, 2022.

[96] Ha, Manh-Hung, and Oscal Tzyh-Chiang Chen. "Deep neural networks using residual fast-slow refined highway and global atomic spatial attention for action recognition and detection." IEEE Access 9 (2021): 164887-164902.

[97] A. Mazari and H. Sahbi. "Coarse-to-fine aggregation for cross-granularity action recognition." In 2020 IEEE International Conference on Image Processing (ICIP), pp. 1541-1545. IEEE, 2020.

[98] Pramono, Rizard Renanda Adhi, Wen-Hsien Fang, and Yie-Tarng Chen. "Relational reasoning for group activity recognition via self-attention augmented conditional random field." IEEE Transactions on Image Processing 30 (2021): 8184-8199.

[99] Li, Jun, Xianglong Liu, Wenxuan Zhang, Mingyuan Zhang, Jingkuan Song, and Nicu Sebe. "Spatio-temporal attention networks for action recognition and detection." IEEE Transactions on Multimedia 22, no. 11 (2020): 2990-3001.

[100] Cai, Jiahui, Jianguo Hu, Shiren Li, Jialing Lin, and Jun Wang. "Combination of temporal-channels correlation information and bilinear feature for action recognition." IET Computer Vision 14, no. 8 (2020): 634-641.

[101] Kusumoseniarto, Raden Hadapiningsyah. "Two-Stream 3D Convolution Attentional Network for Action Recognition." In 2020 Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR), pp. 1-6. IEEE, 2020.

[102] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A.J. Smola and P. Krahenbuhl. Compressed video action recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018), pp. 6026-6035

[103] M. Jiu and H. Sahbi. "Laplacian deep kernel learning for image annotation." In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1551-1555. 2016.

[104] Z. Qiu, T. Yao and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. Proceedings of the IEEE International Conference on Computer Vision (2017), pp. 5533-5541

[105] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018), pp. 6450-6459

[106] L. Wang, W. Li, W. Li and L. Van Gool. Appearance-and-relation networks for video classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018), pp. 1430-1439

[107] M. Zolfaghari, K. Singh and T. Brox. Eco: Efficient convolutional network for online video understanding. Proceedings of the European Conference on Computer Vision (ECCV) (2018), pp. 695-712

[108] L. Wang and H. Sahbi. "Directed acyclic graph kernels for action recognition." In Proceedings of the IEEE International Conference on Computer Vision, pp. 3168-3175. 2013.

[109] J. Lin, C. Gan and S. Han. Tsm: Temporal shift module for efficient video understanding. Proceedings of the IEEE International Conference on Computer Vision (2019), pp. 7083-7093

[110] J Li, Jiapeng, Ping Wei, and Nanning Zheng. "Nesting spatiotemporal attention networks for action recognition." Neurocomputing 459 (2021): 338-348.