Image and Vision Computing 60 (2017) 4–21

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Going deeper into action recognition: A survey☆,☆☆

Samitha Herath*, Mehrtash Harandi, Fatih Porikli

Australian National University, Canberra, Australia Data61/CSIRO, Canberra, Australia

ARTICLE INFO

Article history:
Received 16 May 2016
Received in revised form 14 October 2016
Accepted 25 January 2017
Available online 16 February 2017

Keywords:
Human action recognition
Motion recognition
Survey
Deep networks

ABSTRACT

Understanding human actions in visual data is tied to advances in complementary research areas including object recognition, human dynamics, domain adaptation and semantic segmentation. Over the last decade, human action analysis evolved from earlier schemes that are often limited to controlled environments to nowadays advanced solutions that can learn from millions of videos and apply to almost all daily activities. Given the broad range of applications from video surveillance to human–computer interaction, scientific milestones in action recognition are achieved more rapidly, eventually leading to the demise of what used to be good in a short time. This motivated us to provide a comprehensive review of the notable steps taken towards recognizing human actions. To this end, we start our discussion with the pioneering methods that use handcrafted representations, and then, navigate into the realm of deep learning based approaches. We aim to remain objective throughout this survey, touching upon encouraging improvements as well as inevitable fallbacks, in the hope of raising fresh questions and motivating new research directions for the reader.

1. Introduction

Imagine the time when your smart environment and your robot assistant are capable of recognizing and understanding your actions at a level that they may actually help you in getting things done without your intervention.

We may not be there yet, but our technological progress is geared evidently towards such a marvelous time. In this survey, we walk through existing research on action recognition in the hope of shedding some light on what is available now and what needs to be done in order to develop smart algorithms that are semantically aware of our actions.

1.1. But first, what is an action?

Human motions extend from the simplest movement of a limb to complex joint movement of a group of limbs and body. For instance, while the leg movement on a football kick is a simple motion, jumping for a head-shoot would be a collective movements of legs, arms, head, and whole body. Despite its intuitive and rather simple concept, the term *action* seems to be hard to define! Below, we provide a few examples from the literature:

- Moeslund and Granum [85] and Poppe [98] define *action primitives* as "an atomic movement that can be described at the limb level". Accordingly, the term *action* defines a diverse range of movements, from "simple and primitive ones" to "cyclic body movements". The term *activity* is used to define "a number of subsequent actions", representing a complex movement. For instance, left leg forward is an action primitive of running. Jumping hurdles is an activity performed with the actions starting, running and jumping.
- Turaga et al. [132] define *action* as "simple motion patterns usually executed by a single person and typically lasting for a very short duration (order of tens of seconds)". Their *activity* refers to "a complex sequence of actions performed by several humans who could be interacting with each other in a constrained manner". For example, actions are walking or swimming, activities are two persons shaking hands or a football team scoring a goal.
- Chaaraoui et al. [13] suggest a hierarchical breakdown of human motions in the context of human behavior analysis. The breakdown is based on the level of semantics and the temporal granularity, and considers the "action" in a level between the "motion" and the "activity". Actions are defined as primitive movements (*e.g.*, sitting, walking) that can last up to several minutes.
- Wang et al. [145] suggest that the true meaning of an *action* lies in "the change or transformation an action brings to the environment", *e.g.*, kicking a ball.

In the Oxford Dictionary, *action* is defined as "the fact or process of doing something, typically to achieve an aim" and *activity* is "a thing

☆ This paper has been recommended for acceptance by Jiwen Lu.
☆☆ This work was supported in part by the ARC under Grant DP150104645.
* Corresponding author.
E-mail address: samitha.herath@data61.csiro.au (S. Herath).

http://dx.doi.org/10.1016/j.imavis.2017.01.010
0262-8856/

that a person or group does or has done". We provide a consolidated definition that serves our purposes in this study the best.

> "***Action*** *is the most elementary human*[1]*-surrounding interaction with a meaning.*"

The *meaning* associated with this *interaction* is called the *category* of the *action*. In general, human actions can take various physical forms. In our definition, the term *interactions* can be understood as *relative motions* with respect to the surrounding that may or may not cause a change. In some situations, one may need to associate "surrounding" to particular objects to derive a meaningful interpretation (*e.g.*, brushing hair). This is aligned with the definition of Wang et al. [145], where an action is defined by the change it is brought to the environment.

As an example, consider the motion sequence in Fig. 1. First, consider a primitive leg motion performed by the player on his run. Even though such movement is a relative motion with respect to the surrounding, we can barely attach a meaning to it. On the other hand, the collective motion of limbs, which results in running, has a meaning. Since this is the most elementary and meaningful motion, we consider it as an action, "the running action". Similarly, it is clear that the player's kick and the Jump of the goal-keeper are two distinct actions with labels "kicking" and "jumping".

1.2. Every survey paper has a taxonomy

True, but a generic taxonomy eludes us! Instead, we group solutions based on the fundamental understanding the reader will take at the end. We dedicate a separate section to deep learning based techniques where we discuss various architectures and training methods. At the same time, we arrange video representation based solutions, *i.e.*, methods based on the handcrafted features, at the level of locality that their representations are constructed. We consider that this dual nature of taxonomy is useful in highlighting essential components of the two categories.

To have a glance, the topics that will be covered in our study are shown in Fig. 2.

1.3. Why should we learn more about action recognition?

Analyzing motions and actions has a long history and is attractive to various disciplines including psychology, biology and computer science (see Table 1 for the list of surveys related to motion and action recognition in computer vision). One can trace the fascination about motion back to 500 BC with Zeno's dichotomy paradox. From an engineering perspective, action recognition extends over a broad range of high-impact societal applications, from video surveillance to human–computer interaction, retail analytics, user interface design, learning for robotics, web-video search and retrieval, medical diagnosis, quality-of-life improvement for elderly care, and sports analytics. The long list of emerging technologies and applications (see for example Ahad et al. [2]) points to "manually analyzing action and motion data is impossible".

2. Where to start from?

Let us begin by quoting a visionary thought from early eighties: "*First, there must be a symbolic system for representing the shape information in the brain, and, secondly the brain must contain a set of processors capable of deriving this information from images*" [76]. In the context of action recognition, a good representation must "be easy to compute", "provide description for a sufficiently large class of actions", "reflect the similarity between two like actions", and "be robust to various variations (*e.g.*, view-point, illumination)".

Earliest works in action recognition make use of 3D models to describe actions (See Fig. 3). One notable example is the *WALKER* hierarchical model introduced in Hogg [46] to understand and interpret human actions. Another example is the use of connected cylinders to model limb connections for pedestrian recognition [108].

Generally speaking, constructing accurate 3D models from videos is difficult and expensive. Therefore, many solutions avoid 3D modeling and instead opt for representing actions at a holistic or local level.[2] Formally, we can define:

- *Holistic representations*. Action recognition is based on the extraction of a global representation of human body structure, shape and movements.
- *Local representations*. Action recognition is based on the extraction of local features.

2.1. Holistic representations

We begin by describing the influential work of Bobick and Davis [7]. Motion Energy Image (MEI) and Motion History Image (MHI) are introduced in Bobick and Davis [7]. As the names suggest, the underlying idea is to encode the motion-related information by a single image. The MEI template is a binary image describing where the motion happens and is defined by

$$E_\tau(x,y,t) = \bigcup_{i=0}^{\tau-1} D(x,y,t-i)\ . \tag{1}$$

Here, $D(x,y,t)$ is a binary image sequence representing the detected object pixels while E_τ denotes the formed MEI at a time τ. The MHI template shows how the motion image is moving. Each pixel in MHI is a function of the temporal history of the motion at that point (*i.e.*, higher intensities correspond to more recent movements) (see Fig. 4 for an illustration).

The MEI and MHI templates contain useful information about the context of videos. For example, the gradient of the MHI template is used to filter out the moving and cluttered background in Tian et al. [129]. This is achieved by determining key motion regions in the MHI template using Harris interest point detector [41], followed by identifying the moving/cluttered background as regions with inconsistent motions around the interest points.

The volumetric extension of MEI templates is introduced in Blank et al. [6]. The main idea is to represent an action by a 3D shape induced from its silhouettes in the space–time (see Fig. 5). For classification purposes, the resulting 3D surface is converted to a 2D map by computing the average time each point inside the surface requires to reach the boundary. A related study suggests to represent the MHI templates by spatiotemporal volumes [148], demonstrating that extension to 3D volumes adds robustness to view point variations.

Yilmaz and Shah [157] propose to identify actions based on the differential properties of the *Space–Time Volume* (STV). An STV is built by stacking the object contours along the time axis (see Fig. 5). Changes in direction, speed and shape of an STV inherently characterize the underlying action. Action sketch is a set of properties extracted from the surface of an STV (*e.g.*, Gaussian curvature) and is shown to be robust to view point changes.

[1] In this survey we are chiefly interested in human actions. Nevertheless, an action can be defined in a broader context by excluding the dependency on humans (*e.g.*, actions performed by robots).

[2] Action recognition from Motion Capture Systems (MoCap) and RGBD data is an active line of research these days. Interested reader is referred to the work of Harandi et al. [40], Vemulapalli et al. [137], Koniusz et al. [60], and Rahmani and Mian [101] for the MoCap data and Oreifej and Liu [94], Du et al. [25], Rahmani and Mian [102], and Liu et al. [73] for the RGBD data.

Fig. 1. Actions are "meaningful interactions" between humans and the environment.

2.2. A statistically correct message

Holistic representations flooded the research in action recognition roughly between 1997 and 2007, since such representations are more likely to preserve the spatial and temporal structures of actions. However, nowadays local and deep representations are favored [56,96,115,141]. Various reasons are attributed to this shift. For example, Dollar et al. [20] claim that the holistic approaches are too rigid to capture possible variations of actions (*e.g.*, view point, appearance, occlusions). Matikainen et al. [79] believe that silhouette based representations are not capable of capturing fine details within the silhouette. As such, maybe it is time to change the gear and delve into local and deep solutions!

3. Local representation based approaches

Local representations for action recognition emerge as a result of the seminal work of Laptev [66] on Space–Time Interest Points (STIPs). As in the case of images, local representations for action recognition follow the pipeline of *interest point detection* → *local descriptor extraction* → *aggregation of local descriptors*. Below, we review the key ideas and major developments for the aforementioned components separately.

3.1. Interest point detection

To build an STIP detector, Laptev [66] extends the *Harris* corner detector [41] to *3D-Harris* detector. In 3D-Harris, in addition to rich spatial structures, temporal significance is required to fire the detector. The idea of the 2D Harris corner detector is to find spatial locations in an image with significant changes in two orthogonal directions. The 3D-Harris detector identifies points with large spatial variations and non-constant motions. An example of such requirements is shown in Fig. 6.

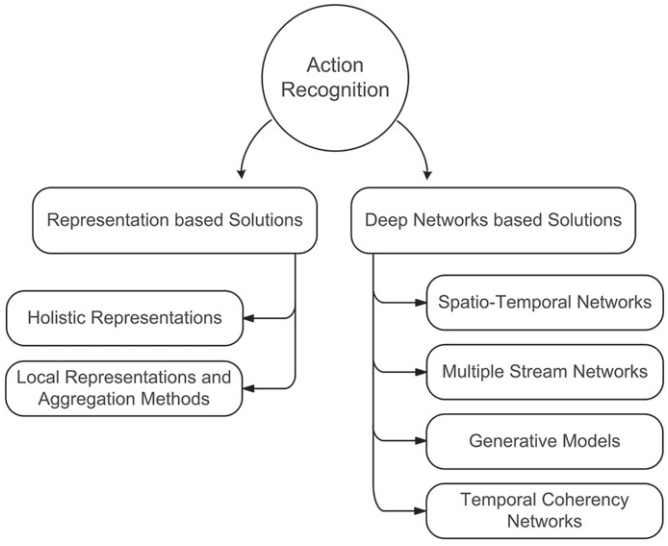

Fig. 2. A taxonomical breakdown we follow in this survey.

Another widely used 2D interest point detector, the *Hessian* detector, is also extended to its 3D counterpart in Willems et al. [150]. Unlike the 3D-Harris detector where gradients are used towards detecting interest points, 3D-Hessian detector makes use of the second order derivatives for its decisions.

In certain domains, *e.g.*, facial expressions, Dollar et al. [20] notice that true spatiotemporal corners, as required by the 3D-Harris or 3D-Hessian detector, are quite rare, even if an interesting motion is occurring. While sparseness is desirable to an extent, STIPs that are too rare can lead to difficulties in action recognition. To overcome this limitation, in Dollar et al. [20] it is proposed to disintegrate spatial filtering from the temporal one. The resulting detector is shown to respond to any region with spatially distinguishing characteristics undergoing a complex motion.

Unlike images, action clips are more likely to be obtained in uncontrolled environments. As such, care should be taken in processing videos since the possibility of good features latching into irrelevant details is high. For example, a shaky camera can fire a series of irrelevant interest points. To address this issue, Liu et al. [72] suggest to prune irrelevant features using statistical properties of the detected interest points. Furthermore, spatiotemporal features obtained from background, known as *static* features, especially the ones that are near motion regions are useful for action recognition [72]. The relevance of static features for action recognition should not sound counter-intuitive. This is because the background in certain types of videos (*e.g.*, football) can provide useful contextual information for action recognition. Moreover and from psychology we know that human beings are able to recognize many types of actions from still images without motion information.

3.2. Local descriptors

Let us start with a simple definition, a 3D cuboid or simply a cuboid is a cube constructed from pixels around detected interest points. To obtain the local descriptor at an interest point, earlier works almost unanimously opt for cuboids [20,66]. In 2009, separate studies by Messing et al. [81] and Matikainen et al. [79] questioned the choice of fixed shaped cuboids for action recognition and introduced the notion of trajectories. Below, we first discuss various local descriptors widely used for action recognition, remembering that

Table 1
Surveys on motion and action analysis.

Survey	Scope
Moeslund and Granum [85]	Human motion capture and analysis
Yilmaz et al. [156]	Object detection and tracking
Turaga et al. [132]	Human actions, complex activities
Zhan et al. [159]	Surveillance and crowd analysis
Poppe [98]	Human action recognition
Weinland et al. [149]	Action recognition
Aggarwal [1]	Motion analysis fundamentals
Chaaraoui et al. [13]	Human behavior analysis and understanding
Metaxas and Zhang [82]	Human gestures to group activities
Vishwakarma and Agrawal [139]	Activity recognition and monitoring

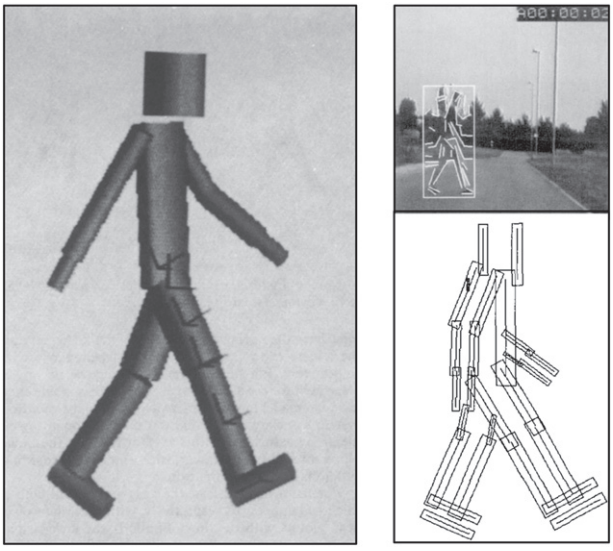

Fig. 3. Early approaches represent actions by 3D models. Left: Hogg [46] introduces the *WALKER* framework to represent walking action using 3D models. The walking pattern is modeled by a sequence of 3D structures. Right: Rohr [108] extended the *WALKER* framework for pedestrian recognition. The model uses connected cylinders and their evolution to identify pedestrians.

local descriptors can be employed with both cuboids and trajectories. We then review trajectories and their improvements.

3.2.1. Edge and motion descriptors

Kläser et al. [58] suggest using the Histogram of Gradient Orientations as a motion descriptor. While being inspired by the Histogram of Oriented Gradients (HoG) [17], the descriptor itself is spanned to the spatiotemporal domain, hence named the *HoG3D* descriptor.

Optical flow fields encode the pixel level motions in a video clip. Exploiting this property, Laptev et al. [67] propose the Histogram of Optical Flow (HoF) over local regions as a spatiotemporal descriptor. A more robust extension of the HoF descriptor is the Motion Boundary Histogram (MBH) introduced in Dalal et al. [18]. MBH is computed over the Motion Boundary fields, *i.e.*, the spatial derivative of optical flow fields (see Fig. 7 for an example). Though being rich, computing optical flow fields is computationally expensive. To overcome this difficulty, Kantorov and Laptev [55] propose to make use of video decompression techniques. More specifically, instead of computing the optical flow fields for obtaining MBH or HoF descriptors, the authors use the motion fields in MPEG compression. This motion field, termed *MPEG Flow*, can be obtained virtually free in the video decoding process.

3.2.2. Pixel pattern descriptors

Local binary patterns (*LBP*) are intensity-based 2D descriptors, successfully used in a diverse range of problems in vision including face recognition and texture analysis [91]. The LBP descriptor is computed by quantizing the neighborhood of a pixel with respect to its intensity. In Zhao and Pietikainen [160], various extensions of the 2D LBP descriptors to spatiotemporal domain are introduced. In the Volume LBP (VLBP), local volumes are encoded by the histogram of the binary patterns [160]. Despite its simplicity, the number of distinct patterns produced by VLBP can become overwhelming for large neighborhoods. To alleviate this difficulty, in the local binary pattern histograms from three orthogonal planes (LBP-TOP), the descriptor is obtained by concatenating local binary patterns on three orthogonal planes, namely *xy*, *xt* and *yt* planes (see Fig. 8 (left) for an illustration for the LPB variant of Kellokumpu et al. [57]). The idea of three orthogonal planes is extended by Norouznezhad et al. [89] to nine symmetric planes.

Describing image regions through second order statistics is proposed in Tuzel et al. [133]. In particular, to describe a region *R* in an

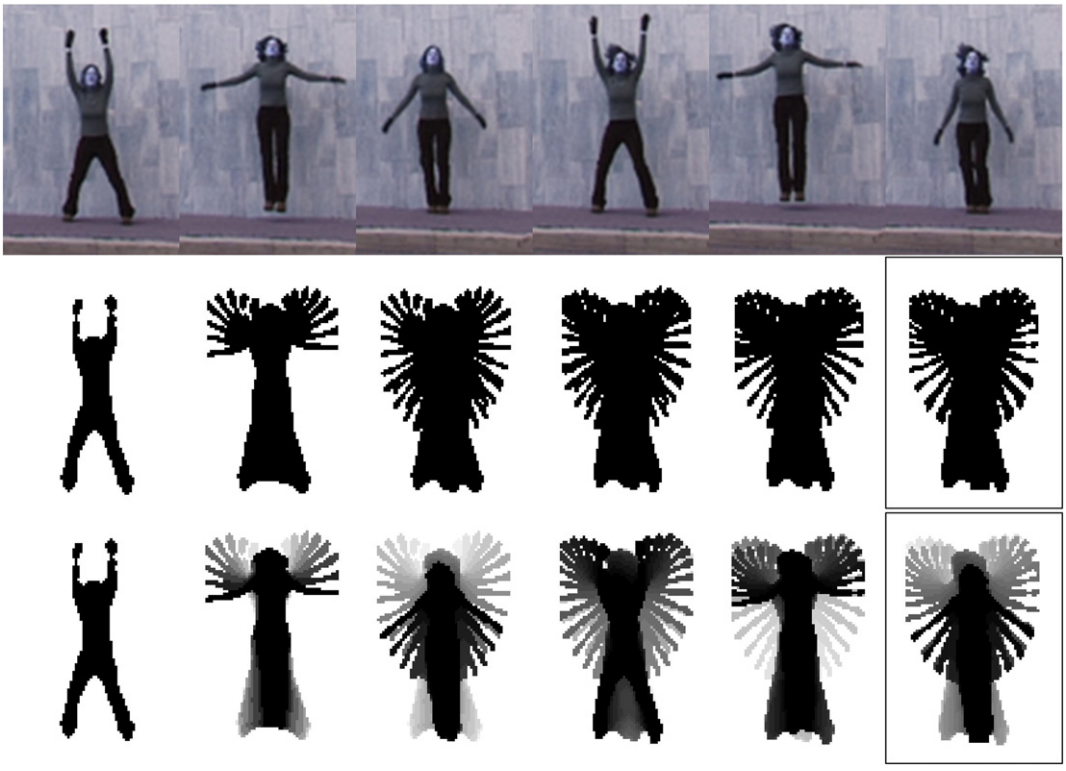

Fig. 4. Top: A jumping sequence. Middle: The MEI template [7]. Bottom: The MHI template [7]. The MEI captures where the motion happens while the MHI template shows how the motion image is moving. The templates at the end of the action, shown in the rightmost column are used for representations.

Fig. 5. Left: The spatiotemporal volumes used by Blank et al. [6] to describe the evolution of an action. The 3D representation is converted to a 2D map by computing the average time taken by a point to reach the boundary. Right: The spatiotemporal surfaces of Yilmaz and Shah [157] for a tennis serve and a walking sequence. The surface geometry (*e.g.*, peaks, valleys) is used to characterize the action.

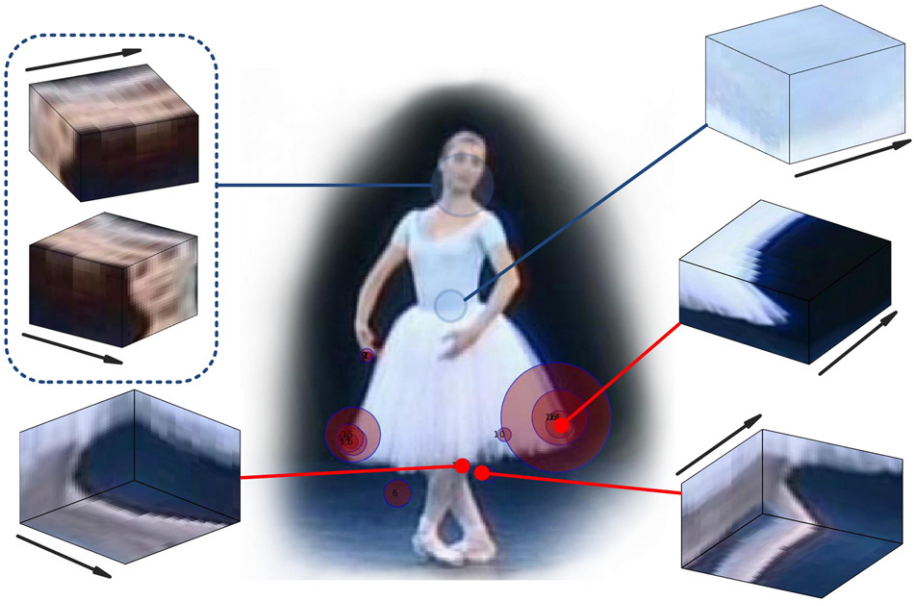

Fig. 6. Marked in red are the detected spatiotemporal interest points of Laptev [66]. Spatial changes along the time axis (marked with an arrow) are noticeable. In this ballet video, the dancer keeps her head still throughout the video. Hence, despite having significant amount of spatial features, no spatiotemporal interest point is detected on the face. Similarly, in her waist no spatiotemporal interest point can be detected as a result of limited spatial variations.

image (see extensions to videos in Sanin et al. [112]), first a set of features $\{\boldsymbol{z}_i\}_{i=1}^{n}, \boldsymbol{z}_i \in \mathbb{R}^d$ is extracted from R (dense or sparse). Common choices here are low-level features (*e.g.*, gradients, RGB intensities) or mid-level features (*e.g.*, SIFT or HoG) [10]. The $d \times d$ covariance matrix of $\{\boldsymbol{z}_i\}_{i=1}^{n}$, usually referred to as Region Covariance Descriptor (RCD), is then used as the descriptor for R. Considering its natural Riemannian structure, RCDs are robust to scale and translation variations, and show resiliency to noise [134] (see Fig. 8 (right) for an illustration).

3.2.3. From cuboids to trajectories

A spatiotemporal interest point might not reside at the exact same spatial location within the temporal extends of a cuboid. Hence, features extracted from cuboids may not necessarily describing the interest point itself. A trajectory is a properly tracked feature over time,[3] (see Fig. 9). Extracting local features from trajectories gains its popularity mostly from the work of Messing et al. [81] and Matikainen et al. [79]. Interestingly, both studies use a form of velocity of trajectories as local features.

Relative motions (*e.g.*, differences in direction, magnitude and location) between trajectories can characterize certain action categories, especially, the categories that involve human/human interactions (*e.g.*, hand-shaking) as shown by Jiang et al. [54]. Rectifying trajectories using camera motions leads to improvements as shown in Jiang et al. [54] and Wang and Schmid [141]. Jiang et al. [54] cluster trajectories to determine the dominant motion in a sequence. The dominant motion is assumed to be caused by the camera and is compensated from original trajectories by subtraction [54] or through affine transformations [50]. Nevertheless, both studies find that the compensation may become misleading if a sizable portion of the video is covered by the actual action. The homography between consecutive frames is also used to estimate the camera motion[4] [141].

3.3. Sparse or dense?

In short, *sparse is old, dense is new!* While early studies opt for sparse interest points, later, several studies show the superiority of dense sampling in both image [28,90] and video classification [142]. A comprehensive comparison between various sparse methods and dense sampling for several descriptors in action recognition can be found in Wang et al. [142].

[3] In 1973 Johansson showed that human subjects could correctly perceive "point-light walkers", a motion stimulus generated by a person walking in the dark, with points of light attached to the its body. This study resembles the notion of trajectories.

[4] We note that Mikolajczyj and Uemura propose to make use of homography for compensating camera motions earlier [83].

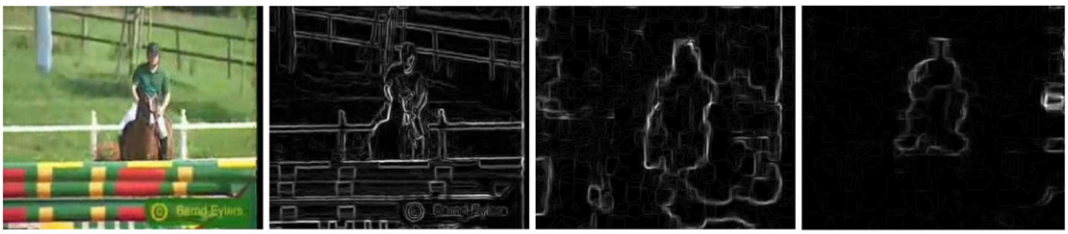

Fig. 7. The spatial gradients (b), horizontal (c) and vertical (d) motion boundary images for the horse riding action in (a). Unlike the spatial gradient which disregards motion information, the motion boundary images stress on the moving object boundaries. Motion boundary images are obtained by computing the gradients of the optical flow fields.

3.4. Aggregation

Let $\mathbb{F} = \{\boldsymbol{f}_i\}_{i=1}^n, \boldsymbol{f}_i \in \mathbb{R}^d$ be a set of local features extracted from a video. For the purpose of action recognition, we need a mechanism to learn from such sets and eventually compare them. Learning algorithms such as Support Vector Machines (SVM) mostly accepts fixed-size vectors and cannot work with sets of varying sizes (the number of local features varies per video). As such and in order to benefit from various learning techniques, we need a mechanism to aggregate sets of local features into discriminative and fixed-size descriptors. In doing so, machineries based on the concept of Bag-of-Visual Words (BoV) [16] and *Dictionary learning* [26,92] are the most natural choices.

3.4.1. Aggregation with BoV

In a nutshell, given a "visual vocabulary" or "codebook" $\mathbb{D} = \{\boldsymbol{d}_j\}_{i=1}^k, \boldsymbol{d}_j \in \mathbb{R}^d$, the distribution of a given set of local descriptors $\mathbb{F} = \{\boldsymbol{f}_i\}_{i=1}^n, \boldsymbol{f}_i \in \mathbb{R}^d$ on the codebook $\mathbb{D}$ is used as the descriptor.

In the BoV, the histogram of "visual word" occurrences is used as the descriptor. That is the frequency of seeing each visual word $\boldsymbol{d}_j$ as the closest match to the local features $\boldsymbol{f}_i$ determines the descriptor. The work of Dollar et al. [20] is among the first studies that resort to BoV for action recognition. In its original form, the temporal information is ignored by BoV. To ameliorate this shortcoming, Laptev et al. [67] propose the spatiotemporal grids. The main idea is to split a video into several sub-videos, aggregate the local descriptors of each sub-video to form the so-called "channels" and compare videos based on their channel descriptors. An improvement inline with the concept of BoV is the hierarchical BoV [61]. The base-level vocabulary is learned using HoG3D descriptors [58]. Other levels of vocabulary are then constructed by aggregating their immediate lower level descriptors while spatiotemporal neighborhoods are taken into account.

More recently, aggregation through the Fisher Vector (FV) encoding [93,96,140] becomes the method of choice. The FV encoding [97] is an aggregation method based on the principle of the Fisher Kernels [49], which combines the benefits of generative and discriminative approaches to pattern classification. Briefly, the key differences between BoV and FV are **I**) BoV employs hard-assignment towards aggregation while FV benefits from soft-assignment, and **II**) if the underlying model of feature generation is assumed to be a Gaussian Mixture Model, BoV only considers the zeroth-order information (occurrences) in aggregation while FV benefits from both first- and second-order statistics. The FV encoding along trajectories delivered the state-of-the-art performances in several studies (see for example Wang et al. [140], Wang and Schmid [141]). Stacked FVs which can be understood as an extension of spatiotemporal grids of Laptev et al. [67] to FVs is introduced in Peng et al. [96]. A detailed analysis of FVs in action recognition is presented in Oneata et al. [93].

FVs are usually very high dimensional [51] and in certain applications redundant. A simplified version of FV, known as Vector of Locally Aggregated Descriptor (VLAD) [3,51], removes the second-order information from the descriptor. As a result, the dimensionality of VLAD descriptors is almost half of FVs. In Jain et al. [50], Xing et al. [154], Kantorov and Laptev [55] and Sun and Nevatia [124] VLAD descriptors obtained from spatiotemporal features are employed for action recognition. A comparison of speed and accuracy of FV against VLAD can be found in Kantorov and Laptev [55] and Wu et al. [152].

3.4.2. Aggregation with spatiotemporal dictionary learning and sparse coding

Sparse coding has become a popular choice in neuroscience, information theory, signal processing, and other related areas [9,23,26,92] in the last decade. By imposing sparsity, it is possible to represent natural signals such as images using only a few non-zero coefficients, *i.e.* as a linear decomposition using a using a few atoms of a suitable dictionary. In computer vision, sparse image representations were originally introduced for modeling spatial receptive fields of cells in the human visual system by Olshausen and Field [92]. Following studies have been shown to deliver notable results for various visual inference tasks, such as face recognition [151], subspace clustering [27] and image restoration [75] to name a few.

For action recognition, Zhu et al. [161] use the principles of sparse coding to aggregate local spatiotemporal features. Using a learned dictionary, they encode HoG3D descriptors obtained from uniformly distributed spatiotemporal cuboids. They obtain the video descriptor

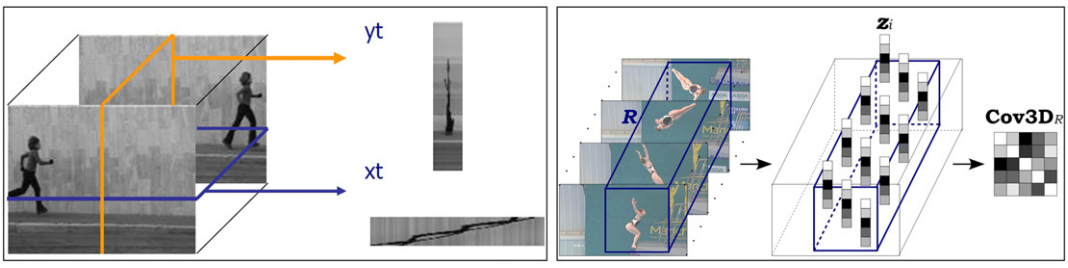

Fig. 8. Left: The LBP extraction planes of Kellokumpu et al. [57] for action recognition inspired by the LBP-TOP descriptor of Zhao and Pietikainen [160]. Here, the video stream is considered as a spatiotemporal volume and LBP descriptors are only extracted from the two orthogonal planes to the image plane Right: The spatiotemporal covariance descriptor of Sanin et al. [112]. Given a spatiotemporal window R, first a set of features $z_i \in \mathbb{R}^d$ is extracted from R (dense or sparse). The spatiotemporal window is then described by the $d \times d$ covariance of the extracted features $z_i \in \mathbb{R}^d$.

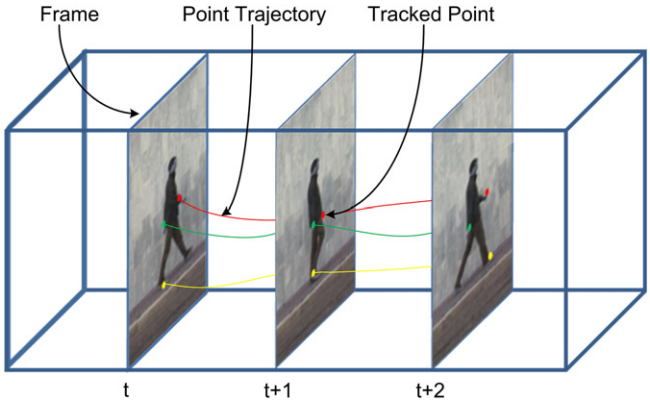

Fig. 9. Tracked point trajectories over frames.

by performing max-pooling on the sparse codes. Moreover, to learn the dictionary, they suggest transfer learning from unlabeled video data.

Guha and Ward [39] study various forms of dictionaries for the task of action recognition. In the simplest form, a common dictionary across all action classes is learned. This common dictionary is shown to be limited in its representative power when new action classes are introduced. To alleviate this limitation, the use of class specific dictionaries is suggested.

To extract spatiotemporal feature, Somasundaram et al. [117] suggest salient spatiotemporal regions. The main idea, inspired by the principals of information theory, states that the saliency of a spatiotemporal region is captured through its structural complexity.[5] Through the use of a dictionary, the structural complexity of a patch is approximated by the concept of *minimum description length* (MDL) [105]. Intuitively, the number of bits required to represent it decreases as regularity in the data increases.

Inspired by the object bank method [70], Sadanand and Corso [111] propose the "action bank", where actions are described by a large set of detectors acting as the dictionary of a high-dimensional "action-space". We point out that *the action bank* itself is a high-level *dictionary*. A relevant idea is presented by Shao et al. [114] where the Laplacian of 3D Gaussian filters is used to construct the action space. Both previous methods exploit the pyramid structure to enhance robustness across spatial and temporal domains.

3.4.3. Aggregation via temporal coherence

We conclude this part by describing studies that explicitly incorporate temporal information in aggregating spatiotemporal information for video descriptors. Fernando et al. [31] propose to represent a video by a ranking machine. That is, given the frame descriptors, a hyperplane that ranks the frames according to their temporal order is used to represent the video. Gaidon et al. [33] propose the concept of atomic-actions or *actoms* which can be understood as a temporally structured extension of the BoV. An actom[6] is the building block of an action and has a variable temporal extend. Histograms of visual words, similar to the traditional BoV approach, are used to describe an actom. Additionally, features that are located in the middle of an actom receive higher weights towards generating the histogram while the contribution of features away from the center is attenuated.

Other notable lines of research that exploit temporal coherence are Hidden Markov Models (HMMs) [100] and Conditional Random Fields (CRFs) [64] that are typically used for sequence modeling applications such as Natural Language Processing [80] and Speech Processing [35]. An action could be interpreted as a sequence of appearance transitions [47,128]. Hence, it is straightforward to model them using a state transition structure such as an HMM.

Hongeng and Nevatia [47] use HMMs in modeling events in videos. However, they break free from the inherent first order dependency between states of HMMs by using the mentioned *semi-HMMs*. The work in Tang et al. [128] employs the max-margin framework for modeling the latent temporal structure of a video. Furthermore, it adopts a *Variable-duration HMM* which additionally associates a latent duration variable along with other latent state variables. In Sun and Nevatia [123], the video is treated as a sequence of short clips corresponding to observations of HMM latent states. The objective of the model is to capture the complex activities by considering actions as such short clips.

In comparison, the discriminatively trained CRF models contain a distinct advantage over their generatively trained counterparts, *i.e.* HMMs. Unlike the first order dependency of HMMs, CRFs are conditioned on the entire sequence. The work of Quattoni et al. [99] embeds latent variables into the CRF modeling and introduces the Hidden CRF (HCRF) for spatiotemporal recognition. In Wang and Mori [147] apply *max-margin* learning in modeling. A hierarchical CRF modeling approach based on the temporal granularity is presented in Song et al. [118].

The work in Li et al. [71] makes the observation that videos usually contain distinct scenes where dynamics are coherent only within them. Therefore, it proposes hierarchical temporal models that are learned on three levels of temporal granularity where the levels are trained using CNN features (we discuss in the next section), linear dynamical systems, and VLAD codes. To capture the nonstationary evolution of dynamics in a video, a hierarchical encoding method is described in Su et al. [122] where they suggest the *Hierarchical Dynamic Parsing and Encoding* pipeline with two or more temporal encoding layers.

We point the reader to the work of Kovashka and Grauman [61], Fernando et al. [30], and Shao et al. [114] for similar ideas on hierarchical video analysis.

4. Deep architectures for action recognition

We are witnessing a significant advancement in countless tasks thanks to the deep and data driven architectures. Deep neural networks such as Convolutional Neural Networks (CNN) [69] have become the method of choice in learning image contents [14,62,126,127]. Generally speaking, the problem of learning is to determine a complicated decision function from the available data. In deep architectures this is achieved by composing multiple level of nonlinear operations. Searching the parameter space of deep architectures is not an easy job given the non-convexity of the decision surface. Learning algorithms based on the gradient descent approach along the computational power of new hardware have been shown to be successful when large amount of annotated data is available [42,121,144].

Our intention in this section is to discuss deep models that have been used (or can potentially be used) to address the problem of learning actions from videos. From a taxonomical point of view, we can identify four categories of architectures applied to action recognition, namely,

- Spatiotemporal networks
- Multiple stream networks
- Deep generative networks
- Temporal coherency networks

Below, we discuss each category in detail and provide pointers to open questions and possible improvements.

[5] We note that a similar concept is used in Laptev [66] to identify spatiotemporal interest points.

[6] A relevant idea to the concept of actoms is proposed by Niebles et al. earlier [88].

4.1. Spatiotemporal networks

The convolutional architecture effectively utilizes the image structure in reducing the search space of the network by "pooling" and "weight-sharing" (see Fig. 10 (left) for a conceptual diagram). Pooling and weight-sharing also contribute to achieving robustness across scale and spatial variations. Analyzing filters learned by CNN architectures suggests that the very first layers learn low level features (*e.g.*, *Gabor*-like filters) while top layers learn high level semantics [158]. This further extends the use of convolutional networks as generic feature extractors.

A direct approach to action recognition using deep networks is to arm the convolutional operation with temporal information. To achieve this, *3D convolutional* networks are introduced in Ji et al. [52]. A 3D convolution network, as the name suggests, uses 3D kernels (filters extended along the time axis) to extract features from both spatial and temporal dimensions, hence is expected to capture spatiotemporal information and motions encoded in adjacent frames (see Fig. 10 for a conceptual diagram). In practice, it is important to provide the network with supplementary information (*e.g.*, optical flow) to facilitate training. Empirically, Ji et al. [52] show that the 3D convolutional networks outperform the 2D frame based counterparts with a noticeable margin.

Generally speaking, the 3D convolutional networks have a very rigid temporal structure. The network accepts a predefined number of frames as the input (for example in Ji et al. [52] the input consists of only 7 frames). While having fixed spatial dimension is somehow defensible (spatial pooling tends to provide robustness across scales), it is unclear why a similar assumption should be made across the temporal domain. Even less clear is the right choice of the temporal span as macro motions in different actions have different speeds and hence different spans.

To answer how temporal information should be fed into convolutional networks, various fusion schemes are investigated. Ng et al. [86] explored temporal pooling and concluded that max pooling in the temporal domain is preferable. Karpathy et al. [56] proposed the concept of *slow fusion* to increase the temporal awareness of a convolutional network. In slow fusion, a convolutional network accepts several, yet consecutive, parts of a video and processes them through the very same set of layers to produce responses across temporal domain. These responses are then processed by fully connected layers to produce the video descriptor (see Fig. 11 for details).

Other forms of fusion include *early fusion* (*e.g.*, the 3D convolutional network [52]) where the network is fed with a set of adjacent frames and *late fusion* where frame-wise features are fused at the last layer [56]. Karpathy et al. [56] also show that a multi-resolutional approach using two separate networks, not only boosts the accuracy but also reduces the number of parameters to be learned. This is due to the fact that each leg of the network (*i.e.*, fovea and context streams in Fig. 11) accepts smaller inputs. We note that the fovea stream receives the central region of a frame to take advantage of the camera bias that exists in many videos since the object of interest often occupies the central region.

Similar to the use of *VGG* [14] and *Decaf* [22] networks as generic descriptors for images, Tran et al. [130] attempt to find generic video descriptors based on a 3D convolutional network. The feature extraction network is trained on Sports-1 M [56] dataset. Empirically, the authors show that a network with 3×3×3 homogeneous filters (constant depth at every layer) performs better than varying the temporal depth on filters. Flexibility on the temporal extent is obtained with the inclusion of 3D pooling layers. A generic descriptor named *C3D*, is then obtained by averaging the outputs of the first fully connected layer of the C3D network.

Varol et al. [136] explore the effect of performing 3D convolutions over longer temporal durations at the input layer. Improvements are observed by extending the temporal depth of the input as well as combining the decision of networks with different temporal awareness at the input.

Extending spatial filters to 3D ones, though being mainstreamed, inevitably increases the number of parameters of the network. In ameliorating the downside effect of 3D filters, Sun et al. [125] suggest factorizing a 3D filter into a combination of 2D and 1D filters. With this reduction of parameters, they obtain comparable performance to Simonyan and Zisserman [115] without any knowledge transfer between several video datasets while training (see Section 4.2 for details on the knowledge transfer of Simonyan and Zisserman [115]).

To exploit the temporal information, some studies resort to the use of recurrent structures. The works of Baccouche et al. [4] and Donahue et al. [21] tackle the problem of action recognition through a cascade of convolutional networks and a class of Recurrent Neural Networks (RNN) [106] known as Long–Short Term Memory (LSTM) [44] networks. As the word *recurrent* suggests, an RNN (see Fig. 12) models the dynamics using a feedback loop. The typical form of an RNN block accepts an external signal $\boldsymbol{x}^{(t)} \in \mathbb{R}^d$ and produces an output $\boldsymbol{z}^{(t)} \in \mathbb{R}^m$ based on its hidden-state $\boldsymbol{h}^{(t)} \in \mathbb{R}^r$ by

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_x \boldsymbol{x}^{(t)} + \boldsymbol{W}_h \boldsymbol{h}^{(t-1)}\right) , \tag{2}$$

$$\boldsymbol{z}^{(t)} = \sigma\left(\boldsymbol{W}_z \boldsymbol{h}^{(t)}\right) . \tag{3}$$

Here, $\boldsymbol{W}_x \in \mathbb{R}^{r\times d}$, $\boldsymbol{W}_h \in \mathbb{R}^{r\times r}$ and $\boldsymbol{W}_z \in \mathbb{R}^{m\times r}$. Obviously, an RNN is a realization of the linear dynamical systems (LDS) [48] and hence rich enough to model video sequences.

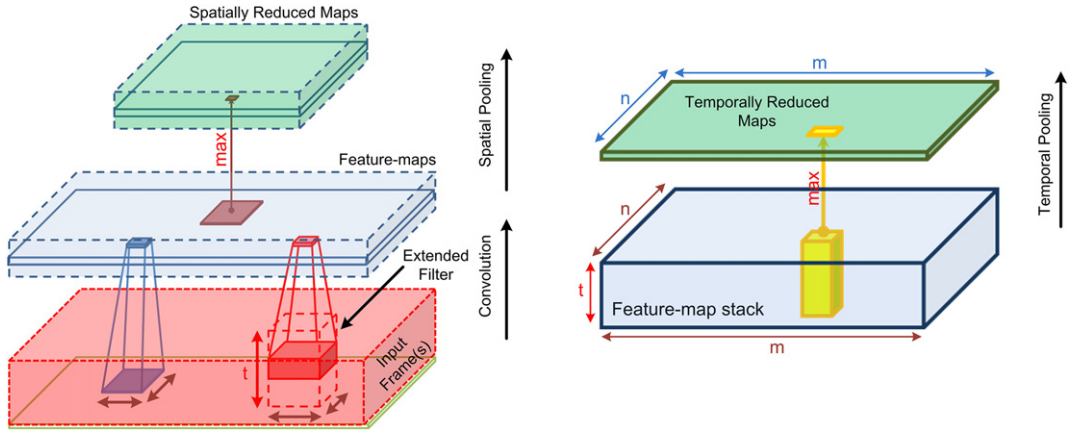

Fig. 10. Spatiotemporal operations: 2D convolution (blue), 3D convolution on frame stacks (red) as in Ji et al. [52], conventional spatial max-pooling (brown), and temporal max-pooling (yellow) as in Ng et al. [86].

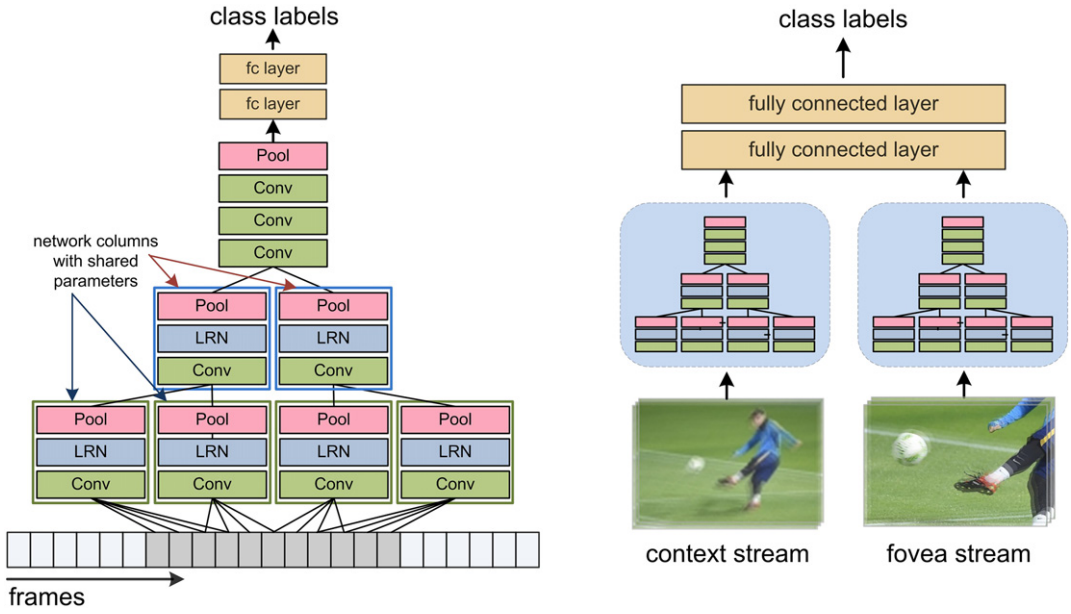

Fig. 11. The foveated architecture of Karpathy et al. [56]. Denoted in green, red and blue are respectively normalization, spatial-pooling and convolutional layers.

Generally speaking, training an RNN is not easy due to the issue of vanishing (or exploding) gradient [5]. For the sake of discussion, assume that the recursive expression of an RNN cell has the form $h^{(t)} = w_h h^{(t-1)}$ with $x, h, z \in \mathbb{R}$. This recursive form can be unfolded as $h^{(t)} = w_h h^{(t-1)} = w_h w_h h^{(t-2)} = \cdots = w_h^t h^{(0)}$. As such, the network either learns short term dependencies (if $w_h < 1$) or very long dependencies (if $w_h > 1$) which is not desirable [5]. LSTM cells (shown in Fig. 12) solve this issue by constraining the states and outputs of the RNN cell through control gates.

To classify actions, Baccouche et al. [4] suggest to feed an LSTM network with features extracted from a 3D convolutional network. The two networks, *i.e.*, 3D convolutional network and the LSTM network are trained separately. That is, first the 3D convolutional network is trained using annotated action data. Once the 3D convolutional network is obtained, the convolutional features are used to train the LSTM network (see Fig. 13 for the network structure).

Another architecture based on LSTM is proposed by Donahue et al. [21] to exploit end-to-end training over the composite network as shown in Fig. 14. The resulting structure named Long-term Recurrent Convolutional Network (LRCN) has been shown to be successful not only in recognizing actions but also in captioning images and videos. With the end-to-end learning and CNN-LSTM convolution, the spatiotemporal receptive filter parameters are computed in a data driven fashion.

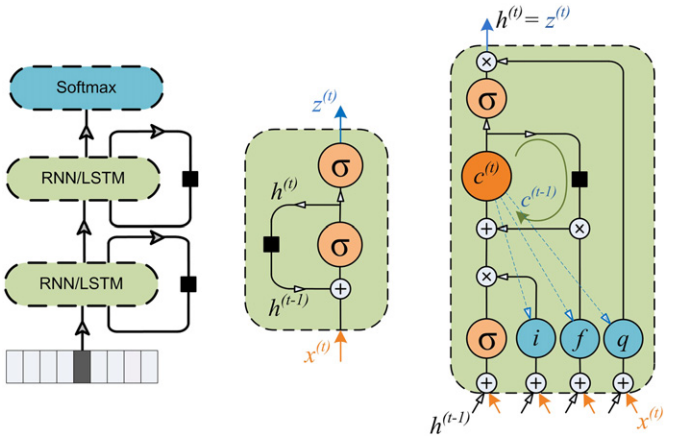

Fig. 12. Left: The recurrent structure of a 2-layer RNN/LSTM network. Center: The RNN cell structure that replicates a linear dynamical system. Right: The LSTM cell that includes additional gate controls. Time delay is indicated with the black square.

4.2. Multiple stream networks

In visual perception, the *Ventral Stream* of our visual cortex processes object attributes such as appearance, color and identity. The motion of an object and its location are handled separately through the *Dosaral Stream* [36]. A class of deep neural networks is devised to separate appearance based information from motion related ones for action recognition [115].

Simonyan and Zisserman [115] introduced one of the first multiple-stream deep convolutional networks where two parallel networks are used for action recognition (see Fig. 14). The so called spatial stream network accepts raw video frames while the temporal stream network gets optical flow fields as input. The following observations are made in Simonyan and Zisserman [115]:

- *Pretraining for the spatial stream network*. Training the spatial stream network from scratch is not the best practice. Empirically, fine-tuning a pretrained network on the ILSVRC-2012 image dataset [110] leads to higher accuracy.
- *Early fusion for the temporal stream network*. Stacking optical flow fields at the input of the temporal stream network (*i.e.*, early fusion) is beneficial.
- *Multi-task learning for the temporal stream network*. The temporal stream network needs to be trained purely from the available video data. This was observed to be challenging for small and medium-size datasets in very deep networks. To circumvent this difficulty, the temporal stream network is modified to have more than one classification layer. Each classification layer operates on a specific dataset (*e.g.*, one operates on the HMDB-51 and one on the UCF-101 dataset) and responds only to the videos coming from the respective dataset. This architecture is a realization of the multi-task learning, aiming to learn a representation, which is not only applicable to the task in question, but also to other tasks.

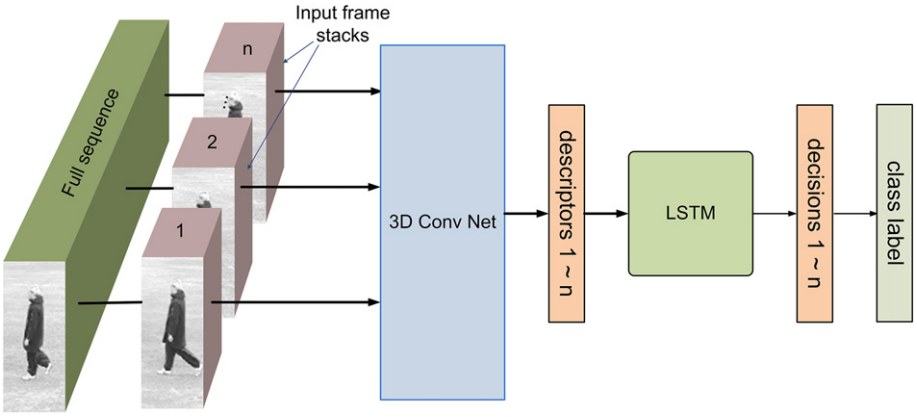

Fig. 13. The network structure of Baccouche et al. [4].

The two streams are fused together using the softmax scores. The work of Feichtenhofer et al. [29] shows that a fusion at an intermediate layer not only improves the performance but also reduces the number of parameters significantly (see Fig. 14 for an illustration). It demonstrates that the best accuracy is attained when the fusion is performed after the last convolutional layer. Interestingly, having the fusion right after the convolutional layers will remove the requirement of costly fully connected layers in both streams. Compared to the original network Simonyan and Zisserman [115], the fused network performs equally well with using only half the parameters.

Extensions of the two stream network include the work of Wang et al. [143] where dense trajectories [141] traced over convolutional feature maps of the two-stream network are aggregated using the Fisher Vector, and [153] where a third stream using audio signal is added to the network.

The optical flow frames are the only motion related information used in two stream networks. This would raise the question whether two stream networks can capture subtle but long-term motion dynamics (such motions cannot be modeled by optical flow). The improvements brought by effective combination of deep architectures and handcrafted solutions hint that certain details in actions are still out-of-reach in deep solutions [29].

4.3. Deep generative models

The potential reward of devising deep models that require little or no supervision is beyond imagination, given the vast and ever increasing videos available on the Web. A good generative model is the one that can learn the underlying distribution of data accurately. Generative models for sequence analysis [120,126] are mainly used to predict the future of a sequence. That is, given a sequence $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_t\}$, one may deem to learn a model to predict its future (*e.g.*, the next instance $\boldsymbol{x}_{t+1}$). This task is different from methods discussed in Section 4.1 in nature as it does not require labels for training. However, accurate predictions are achieved if contents and dynamics (*e.g.*, motion primitives) of the sequence can be captured by the model to a good extent. Deep-generative architectures [37,44,138] aim this goal, *i.e.*, learning from temporal data in an unsupervised matter. In video analysis where annotating data is costly, unsupervised techniques are preferred over supervised ones.

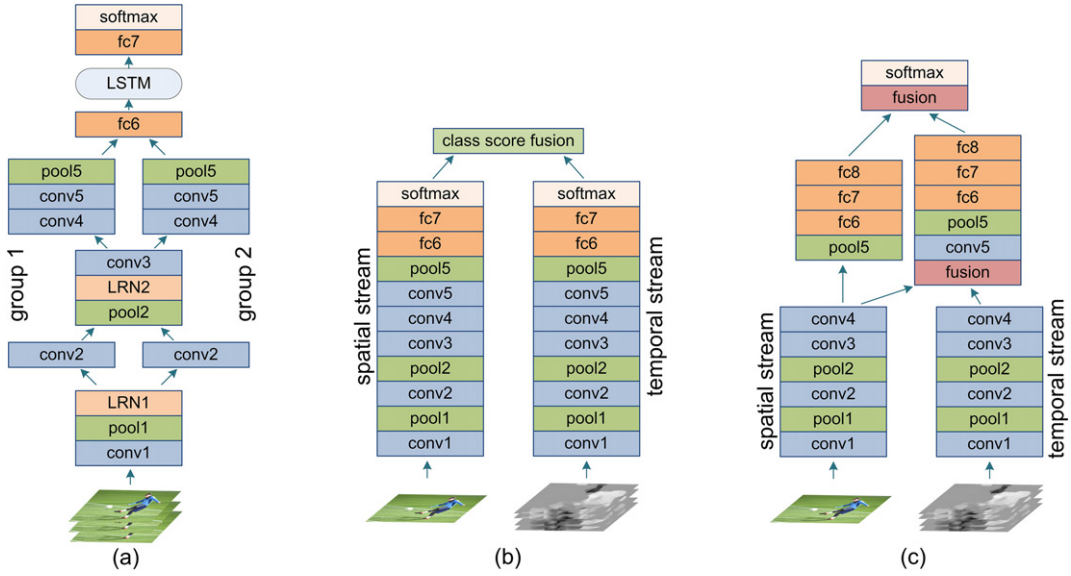

Fig. 14. (a) LRCN network structure of Donahue et al. [21]. A *group* is a set of convolutional filters operating only on a particular set of feature maps from the previous layer. For clarity, we denote each group by separate convolutional blocks. (b) The two-stream network by Simonyan and Zisserman [115] with RGB and stacked optical-flow frames as inputs. (c) An example of a two stream fusion network of Feichtenhofer et al. [29].

Envisaging possible potentials, in this part we review notable examples of deep generative architectures without confiding ourselves to studies that have been directly applied to action recognition.

4.3.1. Dynencoder

Inspired by LDS modeling [24], Yan et al. [155] introduce *Dynencoder*, a class of deep auto-encoders, to capture video dynamics. In its most basic form, a Dynencoder constitutes of three layers. The first layer maps the input $\boldsymbol{x}_t$ to the hidden states $\boldsymbol{h}_t$. The second layer is a prediction layer that predicts the next hidden states, $\tilde{\boldsymbol{h}}_{t+1}$, using current ones (*i.e.*, $\boldsymbol{h}_t$). The final layer is a mapping from the predicted hidden states $\tilde{\boldsymbol{h}}_{t+1}$ to generating estimated input frames $\tilde{\boldsymbol{x}}_{t+1}$. To reduce the training complexity, the parameters of the network are learned in two stages. In the pretraining stage, each layer is trained separately. Once pretraining is completed, an end-to-end fine tuning is performed.

Dynencoder is shown to be successful in synthesizing dynamic textures. One can think of a Dynencoder as a compact way of representing the spatiotemporal information of a video. As such, the reconstruction error of a video given a Dynencoder can be used as a mean for classification.

4.3.2. LSTM autoencoder model

Generative models for action recognition are expected to discover long-term cues, making deep models with LSTM cells natural choices. To this end, Srivastava et al. [120] introduced the LSTM autoencoder model as illustrated in Fig. 15. The LSTM autoencoder consists of two RNNs, namely the *encoder* LSTM and the *decoder* LSTM. The encoder LSTM accepts a sequence (as input) and learns the corresponding compact representation. The states of the encoder LSTM contain the appearance and dynamics of the sequence. As such, the compact representation of a sequence is chosen to be the states of the encoder LSTM. The decoder LSTM receives the learned representation to reconstruct the input sequence. For more details, see Fig. 15.

The LSTM autoencoder can be used to predict the future of a sequence as well. In practice, a composite model that both reconstructs the input sequence and predicts its future delivers the most accurate responses.

4.3.3. Adversarial models

To sidestep various difficulties in training deep generative models, Goodfellow et al. [37] introduce the adversarial networks where a generative model competes with a discriminative model known as an adversary. The discriminative model learns to determine whether a sample is coming from the generative model or the data itself. During training, the generative model learns to generate samples that share more similarities to the original data, while adversary model improves its judgments on whether a given sample is authentic or not. Mathieu et al. [78] adopt the adversarial methodology to train a multi-scale convolutional network for video prediction. They exploit adversarial training to have convolutional networks that avoid pooling layers. They also provide a discussion on the advantages of pooling in generative models.

4.4. Temporal coherency networks

Before concluding this part, we would like to bring the notion of *temporal coherency* into perspective. Temporal coherency is a form of weak supervision and states that consecutive video frames are correlated both semantically and dynamically (*i.e.*, abrupt motions are less likely). For actions, even stronger connections between spatial and temporal cues exist [101]. A sequence is called coherent if its frames are in the correct temporal order. Temporal coherency can be learned by a deep model if the model is fed by ordered and disordered sequences as positive and negative samples, respectively. This concept has been used by Goroshin et al. [38] and Wang and Gupta [146] to learn robust visual representations from unlabeled videos.

Misra et al. [84] study how temporal coherency can be used to train deep models for action recognition and pose estimation. In particular, a *Siamese Network*[15,74,135] (see Fig. 16) is trained with tuples to determine whether a given sequence is coherent or not. Empirically, it has been shown that

- Compared to other supervised pretrained methods, *e.g.*, ImageNet [110], learning by tuples gives more attention to human poses.
- Selection of tuples in the frames with rich motions will avoid ambiguities between positive and negative tuples.

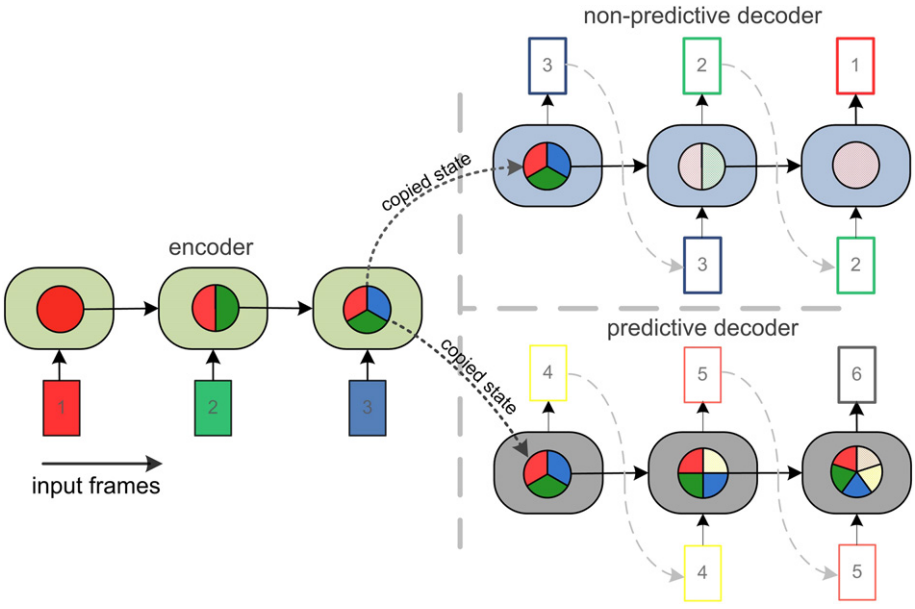

Fig. 15. The composite generative LSTM model by Srivastava et al. [120]. The internal states (represented by the circle inside) of the encoder LSTM capture a compressed version of the input sequence (*e.g.*, frames 1, 2 and 3). The states thereafter are copied into two decoder models, which are reconstructive and predictive. The reconstruction decoder attempts to reconstruct original frames in the reverse order. The predictive model is trained on predicting the future frames 4, 5 and 6. The colors on the state markers indicate the presence of information from a particular frame.

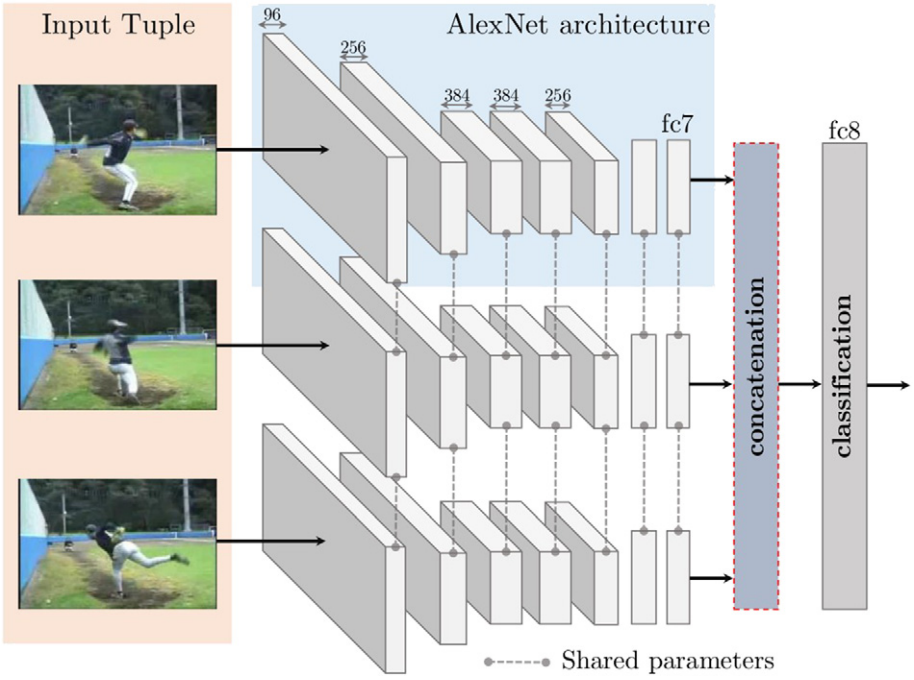

Fig. 16. The Siamese Triplet network used by Misra et al. [84]. Each network is expected to capture the motion and pose of actions.

- Compared to networks trained from scratch, pretrained networks based on the temporal coherency have potential to improve the accuracy.

We note that the temporal coherency is not always a strong assumption to rely on. For example, an abrupt scene variation such as an advertisement shown during an sport event (*e.g.*, in SPORTS-1 M data) can violate the temporal coherency easily [71].

Another related study to temporal coherency is the work of Wang et al. [145] where an action is split into two phases for classification. More specifically, a video with frames $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ is split into the precondition set $\boldsymbol{X}_p = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_p\}$ and effect set $\boldsymbol{X}_e = \{\boldsymbol{x}_e, \boldsymbol{x}_{e+1}, \cdots, \boldsymbol{x}_n\}$. The cardinality of both sets is learned by the deep model. An action is then identified by the transformation required to map a high-level descriptor extracted from $\boldsymbol{X}_p$ to a high-level descriptor extracted from $\boldsymbol{X}_e$. In particular, the high-level descriptor and transformations are learned using the Siamese Networks (see Fig. 17 for details).

Rank pooling [31] is an effective solution for capturing temporal evolution of a sequence. In its original form, learning of video representations (through ranking) and action classification is done separately. This is due to the fact that, unlike other pooling operations such as max pooling, a closed form solution for the rank pooling operation is not readily available. Recently, Fernando and Gould [32] suggest an end-to-end learning scheme that learns both the pooling operation and the classifier with back propagation. A related work, though not being a deep learning based solution, is the *hierarchical* rank pooling [30] that aims to encode multiple levels of dynamical granularity in videos by iteratively applying the rank pooling operations.

For completeness, we conclude this section by discussing the work of Ranzato et al. [103]. Ranzato et al. [103] state that the success of language modeling through RNNs is a result of the discrete information space. Based on this, they introduce a discrete structure for video frames by quantizing them with a representative collection of image patches. Not surprisingly, natural videos seem to lack the

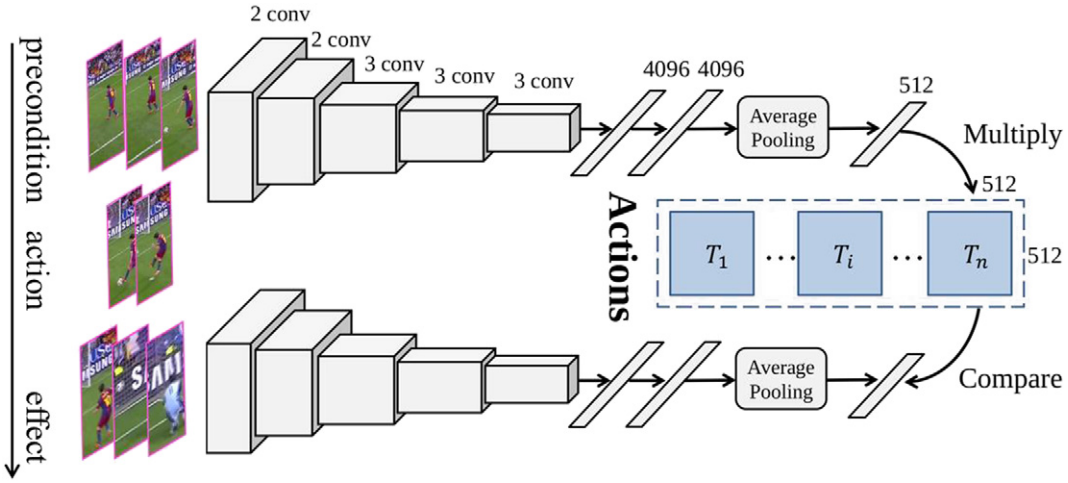

Fig. 17. The parallel convolutional structures are used in extraction of precondition and post-effect features.

dynamics word sequences possess, hinting why language models are superior to their video counterparts. Based on their observations, Ranzato et al. [103] suggest that training a recurrent convolutional network to predict long sequences may lead to better robustness in video modeling.

5. A quantitative analysis

In this section we provide a high level analysis of the aforementioned solutions. We highlight the performance of some notable examples (see Table 3) in our discussion. Having our focus on performances, we also discuss challenges to be addressed and possible directions future solutions may take.

5.1. What is measured by action datasets?

Performance of the action recognition methods is usually compared on a few publicly available datasets. A comprehensive list of available datasets along their details is given in Table 2. In general, there is no such thing as a "universal solution", *i.e.*, a solution that can be applied to any given dataset. As such and if possible, we provide pointers as to why a solution is (un)successful in a scenario.

Along the progress of solutions, action datasets are also evolved in terms of their complexities. The complexity of a dataset is usually described by the resemblance of its contents to reality. For example, the KTH and Weizmann datasets listed on the top of Table 2 contain human actions in controlled conditions (*e.g.*, limited camera motion, almost zero background clutter). Furthermore, their scope is limited to basic actions such as walking, running and jumping. Hence, comparing solutions on the KTH and Weizmann datasets is less insightful unless a specific need is considered. Having said this, we acknowledge that both datasets are useful if motion patterns are considered.

With the increasing complexity, we get to datasets that are composed of YouTube videos, movies and television broadcast snippets (*e.g.*, HMDB-51, UCF-101). YouTube videos are mostly recorded by nonprofessionals with handycams. As a result, they contain camera motion (and shakes), viewpoint variations and resolution inconsistencies. To perform well, a solution must compensate the aforementioned variations. One can observe that in the HMDB-51 and UCF-101 datasets, the actions are well cropped in the temporal domain. Therefore, these datasets are not well-suited for measuring the performance of action localization. An interesting feature of the HMDB-51 and UCF-101 datasets is the inclusion of what we would like to call subtle classes. Examples are *chewing* and *talking* or *playing violin* and *playing cello*. Learning to distinguish between subtle classes requires a deeper understanding of spatial and temporal clues.

Movies and many sports broadcasts are filmed from several viewpoints, and then edited into one stream. This brings sudden viewpoint variations to the video streams. Both Hollywood2 and Sports-1 M datasets contain view-point/editing complexities. Furthermore, the actions usually occur in a small portion of the clip. To make the recognition more challenging, SPORTS-1 M dataset also contains scenes of spectators and banner adverts. Therefore, methods that rely on temporal coherency may fail on Sports-1 M. We note that both SPORTS-1 M and Hollywood2 datasets are specifically annotated by text and script analysis albeit labeling is noisy [56,67].

As mentioned earlier, all action classes in the HMDB-51, UCF-101, Hollywood2 and Sports-1 M datasets *cannot* be distinguished by motion clues. In such situations, the objects contributed to the actions become important as certain actions are defined by the related objects. A good example for this is the 23 distinct types of billiard categories given in Sports-1 M. Hence, algorithms benefiting from object details are expected to perform better.

Deep architectures are notorious for their data-hungry nature. As such, tuning deep networks on small and medium size action datasets such as KTH and Weizmann is difficult and often leads to unsatisfactory performance [125]. The Sports-1 M dataset is assembled to alleviate this limitation, making training and tuning very deep networks possible.

5.2. Recognition results

In Table 3, we provide a comprehensive list of 31 must-know methods along their accuracies on seven challenging action datasets. The accuracies are reported directly from the original works. Instead

Table 2
Datasets for action recognition.

Dataset	Source	No. of videos	Video duration	Training protocol	No. of classes	Videos/class	Example classes
KTH [113]	Recoded videos on both outdoors and indoors	600	4 s	Training and testing are divided on subjects	6	–	Walk, jog, run
Weizmann [6]	Outdoor video recordings on still backgrounds	90		Leave out one cross validation	10	–	Walk, jump, jumping jack, skip
UCF-Sports [107]	Television sports broadcasts (*e.g.* BBC, ESPN) (780 × 480)	150	6.39 s	Classification accuracy on provided train test splits by Tian Yan, discriminative figure-centric models for	10	6–22	Diving, golf-swing, kicking
Hollywood2 [77]	Clips from 69 Hollywood movies (33 training and 36 testing) annotated based on movie script	1707		mAP of each class (884 test videos and 823 training videos obtained from separate training and testing movies)	12	20–140	Answer-phone, eat, handshake
Olympic Sports [88]	YouTube video sequences			mAP of each class on provided train–test splits	16	50	High-jump, long-jump, triple-jump
HMDB-51 [63]	YouTube, movies	7000	2–3 s	Classification accuracy of 30 test clips with training on 70 clips (3 splits are provided)	51	Over 101	Brush-hair, kick, kiss
UCF-50 [104]	YouTube video sequences	–	–	Leave out one cross validation	50	–	Rowing, fencing, punch
UCF-101 [119]	YouTube video sequences	13,320	2–5 s	Classification accuracy on 3 train and test spits	101	Over 100	Diving, skiing, apply eye makeup
Sports-1 M [56]	YouTube sports videos annotated automatically from YouTube topics	1,133,158		70% of as training while testing and validation sets are respectively 20% and 10%.	487	1000–3000	Cricket, disc golf, gliding

Table 3
Accuracy of action recognition techniques (numbers are true recognition accuracy given in percentages). The column *Type* indicates whether a method is purely Deep-net based (D), representation based (R) or fused solution (F).

Reported paper	Method	Type	Dataset						
			HMDB51	UCF101	UCF50	UCF-Sports[a]	Hollywood2[a]	Olympic Sports[a]	Sports-1 M
Wang et al. [140]	Dense Traj (Traj + HoG + HoF + MBH)	R				88.2	58.3		
Kliper-Gross et al. [59]	Motion interchange patterns	R	29.2		68.5				
	General		26.9						
Sadanand and Corso [111]	Video wise	R			76.4				
	Group wise				57.9				
Oneata et al. [93]	MBH + SIFT + Sqrt + L2 normalization	R	54.8		90		63.3	82.1	
	Without human detector	R	55.9		90.5		63	90.2	
Wang and Schmid [141]	With human detector		57.2		91.2		64.3	91.1	
Jain et al. [50]	Traj + HoG + HoF + MBH + DCS on *w*-flow	R	52.1				62.5		
Peng et al. [96]	Stacked FVs + FV	R	66.8						
Peng et al. [95]	Hybrid-BoW	R	61.1	87.9	92.3				
Kantorov and Laptev [55]	MPEG-flow: VLAD encodings of	R	46.3						
Gaidon et al. [34]	SDT tree ATEP	R	41.3				54.4	85.5	
Simonyan and Zisserman [115]	Two-stream (CNN-M-2048)	D	59.4	88.0					
	Transfer learning on Sports-1 M			65.4					
Karpathy et al. [56]	Clip hit @ 1 — slow fusion	D							41.9
	Video hit @ 1 — slow fusion								60.9
Sun et al. [125]	Factorized spatiotemporal conv. nets	D	59.1	88.1					
	Two-stream (ClarifaiNet)			88.0					
Wang et al. [144]	Two-stream (GoogLeNet)	D		89. 3					
	Two-stream (VGG-16)			91.4					
Wang et al. [143]	TDD + Wang and Schmid [141]	F	65.9	91.5					
	TDD (only)	F	63.2	90.3					
	Conv pooling hit @ 1 (best)								72.4
Ng et al. [86]	LSTM hit @ 1 (best)	D							73.1
	Conv pooling (image + opt flow)			88.2					
	LSTM (image + opt flow)			88.6					
Fernando et al. [31]	Rank pooling	R	63.7				73.7		
Donahue et al. [21]	LRCN-weighted average of RBG + flow	R		82.9					
Wu et al. [153]	Adaptive multi-stream fusion	D		92.6					
Jiang et al. [53]	TrajShape + TrajMF	R	48.4	78.5			55.2	80.6	
	TrajShape + TrajMF + Wang and Schmid [141]		57.3	87.2			65.4	91	
Lan et al. [65]	Multi-skip feat. stacking	R	65.1	89.1	94.4		68.0	91.4	
Hoai and Zisserman [43]	Proposed SSD + RCS	R	62.2				72.7		
Tran et al. [130]	C3D on SVM	D		85.2					
	C3D + Wang and Schmid [141] on SVM	F		90.4					
Misra et al. [84]	ImageNet pretrain + tuple verification	D	29.9						
	HMDB + UCF101 labels only		30.6						
Wang et al. [145]	Proposed only (RBG + opt flow networks)	D	62	92.4					
Fernando and Gould [32]	End to end rank-pooling	D				87	40.6		
Fernando et al. [30]	Hierarchical rank-pooling (CNN features)	D	47.5	78.8			56.8		
	Hierarchical RP on CNN + Fernando et al. [31]	F	65.0	90.7			74.1		
Li et al. [71]	VLAD^3	F		84.7				90.8	
	VLAD^3 + Wang and Schmid [141]	F		92.2				96.6	
Varol et al. [136]	LTC$_{flow+RGB}$	D	64.8	91.7					
	LTC$_{flow+RGB}$ + Wang and Schmid [141]	F	67.2	92.7					
Feichtenhofer et al. [29]	Two stream fusion (VGG-16)	D	65.4	92.5					
	Two stream fusion (VGG-16) + Wang and Schmid [141]	F	69.2	93.5					
de Souza et al. [19]	Hybrid fusion of Wang and Schmid [141] + Deep-nets	F	70.4	92.5			72.6		

[a] Datasets in which the mean average precision is reported.

of considering each case individually, we opt to give a high level comparison between various classes of solutions.

5.2.1. Almost equal performance?

A quick look at the accuracy scores shows that the state-of-the-art solutions based on both representation and deep learning perform equally well. This would have been an unexpected observation if image classification is considered. For example, the stacked FV encodings of trajectory descriptors [96] outperform the state-of-the-art deep learning based solutions [29,136] on HMDB-51 without trajectory decision fusion. In contrast, the gap between similar solutions (*e.g.*, comparison between SIFT + FV and CNNs in Krizhevsky et al. [62]) is contradictory when it comes to image classification. Among various reasons one can think of, the insufficiency of data cannot be disregarded.[7] A dominant theme to get around this limitation is to benefit from models pre-trained on images [29,115,136].

5.2.2. State-of-the-art solutions

5.2.2.1. Handcrafted solutions. Focusing on the handcrafted solutions, a performance milestone is achieved by the introduction of *dense trajectory descriptors* [141]. The descriptor can be easily incorporated in various pooling strategies such as *FV*'s [96] and *Rank-Pooling* [31], leading to competitive results on the HMDB-51 and UCF-101 datasets.

[7] While ImageNet contains over 1000 training instances per class, HMDB-51 has only around 70.

5.2.2.2. Deep-net solutions. Turning our attention to deep solutions, we find that the spatiotemporal networks [56,130,136] and two-stream networks [29,115] outperform other network structures. Lately, both of these structures are equipped with 3D convolution filters. Examples include the use of 3D convolutions and pooling in Feichtenhofer et al. [29] and Varol et al. [136]. The work in Wang et al. [144] also suggests that deeper models help boosting the performances. However, training deeper networks demands more rigorous data augmentation techniques (*e.g.*, temporal crops by random clip sampling, frame skipping).

5.2.3. *Fusion with dense trajectories seems to always help*

The recognition accuracy (*see* Table 3) of most the state-of-the-art deep learning based solutions [29,130,136] can be improved based on the observations made in Wang and Schmid [141].[8] This indicates that the structures learned by deep networks are complementary to the *handcrafted* trajectory descriptors. It is worth mentioning that both deep networks (in most cases) and trajectory descriptors consider similar inputs (*i.e.*, RGB and optical flow frames). In Simonyan and Zisserman [115], it is observed that some filters in the temporal stream respond to spatial gradients. Similarly, the MBH trajectory descriptor [141] is also derived using spatial gradients on the optical flow frames.

5.3. What algorithmic changes to expect in future?

Following the trend of other developments in computer vision, moving towards deep architectures for action recognition is dominating the action recognition research lately. Given the difficulty of training deep networks when it comes to video data, knowledge transfer, *i.e.* benefiting from models trained on images or other sources, is an avenue to explore. A related and less investigated problem for knowledge transfer in deep networks is the idea of *heterogeneous domain adaptation*[45,131].

Considering deep architectures for action recognition, the keywords to remember would be 3D convolutions, temporal pooling, optical flow frames, and LSTMs. Though the aforementioned elements are developed individually, novel methods aim at blending them to boost the performance [21,29,136]. We consider that this might be an indication of convergence towards a generic form of deep architectures for spatiotemporal learning.

Another point to remember is that, to boost the performance, carefully engineered approaches are needed. For instance, data augmentation techniques [144], foveated architecture [56] and distinct frame sampling strategies [29,115] have been shown to be essential.

5.4. Bringing action recognition into life

Action recognition has advanced from recognition in controlled environments [6,7,108] to solutions that target more realistic activities (see Table 3). However, in order to use these solutions in real-life scenarios, deeper understanding in the following areas is required:

- Action recognition for a practical applications involves joint detection and recognition from a sequence. Some recent works address joint segmentation and recognition of actions [8,11,12].
- Rather than recognizing actions from a big pool of classes, constraining into a refined set of actions can be useful in practical applications (*e.g.*, cooking activities of Rohrbach et al. [109]). Therefore, fine-grained action recognition tasks [68,87,116] that have been already receiving growing attention from the community, can shape the future solutions and associated problems.

[8] In addition to the decision level fusion, the work of Wang et al. [143] and de Souza et al. [19] suggests employing hybrid fusion methods.

6. Conclusion

Despite having similarities to *static image analysis*, video data analysis is far more complicated. A successful video analytic solution not only needs to overcome variations such as scale, intra-class diversities and noise, but also has to analyze motion cues in videos.

Human action recognition can be considered as the queen of video analysis problems due to its wide applications and the complexity of the motion patterns produced by articulated body movements. In this survey, we investigate several aspects of the existing solutions for action recognition. We first review methods based on the handcrafted representations, and then focus on solutions that benefit from deep architectures. We provide a comparative analysis of these two prevailing lines of research.

Acknowledgments

We would like to thank our reviewers for pointing out the ways we could improve this survey. Further, we would like to thank *Dr. Anoop Cherian* and *Dr. Basura Fernando* for fruitful discussions and encouragement comments given for this work.

References

[1] J.K. Aggarwal, Distributed Video Sensor Networks, Springer London, London, 2011, 27–39. Ch. Motion Analysis: Past, Present and Future
[2] M.A.R. Ahad, J.K. Tan, H. Kim, S. Ishikawa, Motion history image: its variants and applications, Mach. Vis. Appl. 23 (2012).
[3] R. Arandjelovic, A. Zisserman, All About VLAD, Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, June 2013. pp. 1578–1585.
[4] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, A. Baskurt, Sequential deep learning for human action recognition, Proceedings of the Second International Conference on Human Behavior Understanding. HBU'11, 2011. pp. 29–39.
[5] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. 5 (2) (1994) 157–166.
[6] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space–time shapes, Proc. Int. Conference on Computer Vision (ICCV), vol. 2, 2005. pp. 1395–1402.
[7] A.F. Bobick, J.W. Davis, The recognition of human movement using temporal templates, IEEE Trans. Pattern Anal. Mach. Intell. 23 (3) (2001) 257–267.
[8] E.Z. Borzeshi, O.P. Concha, R.Y.D. Xu, M. Piccardi, Joint action segmentation and classification by an extended hidden Markov model, IEEE Signal Process Lett. 20 (12) (2013) 1207–1210.
[9] E.J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theory 52 (2) (2006) 489–509.
[10] J. Carreira, R. Caseiro, J. Batista, C. Sminchisescu, Free-form region description with second-order pooling, IEEE Trans. Pattern Anal. Mach. Intell. 37 (6) (June 2015) 1177–1189.
[11] J. Carvajal, C. McCool, B.C. Lovell, C. Sanderson, Joint Recognition and Segmentation of Actions via Probabilistic Integration of Spatio-Temporal Fisher Vectors, 2016. CoRR abs/1602.01601
[12] J. Carvajal, C. Sanderson, C. McCool, B.C. Lovell, Multi-action recognition via stochastic modelling of optical flow and gradients, Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, MLSDA'14, 2014. pp. 19:19–19:24.
[13] A.A. Chaaraoui, P. Climent-Pérez, F. Flórez-Revuelta, A review on vision techniques applied to human behaviour analysis for ambient-assisted living, Expert Syst. Appl. 39 (12) (2012) 10873–10888.
[14] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets, British Machine Vision Conference, 2014.
[15] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '05, 2005. pp. 539–546.
[16] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, In Workshop on Statistical Learning in Computer Vision, ECCV, 2004. pp. 1–22.
[17] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, 2005. pp. 886–893. vol. 1.
[18] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, Proc. European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science (LNCS), vol. 3952, 2006. pp. 428–441.
[19] C.R. de Souza, A. Gaidon, E. Vig, A.M. López, Sympathy for the details: dense trajectories and hybrid classification architectures for action recognition, Proc. European Conference on Computer Vision (ECCV), 2016. pp. 697–716.

[20] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, Proceedings of the 14th International Conference on Computer Communications and Networks, 2005. pp. 65–72.

[21] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. pp. 2625–2634.

[22] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: a deep convolutional activation feature for generic visual recognition, International Conference in Machine Learning (ICML), 2014.

[23] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.

[24] G. Doretto, A. Chiuso, Y.N. Wu, S. Soatto, Dynamic textures, Int. J. Comput. Vis. 51 (2) (2003) 91–109.

[25] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015. pp. 1110–1118.

[26] M. Elad, Sparse and Redundant Representations — From Theory to Applications in Signal and Image Processing, Springer. 2010.

[27] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.

[28] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2005. pp. 524–531. vol. 2.

[29] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. pp. 1933–1941.

[30] B. Fernando, P. Anderson, M. Hutter, S. Gould, Discriminative hierarchical rank pooling for activity recognition, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[31] B. Fernando, E. Gavves, M.J. Oramas, A. Ghodrati, T. Tuytelaars, Modeling video evolution for action recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. pp. 5378–5387.

[32] B. Fernando, S. Gould, Learning end-to-end video classification with rank-pooling, ICML, 2016.

[33] A. Gaidon, Z. Harchaoui, C. Schmid, Actom sequence models for efficient action detection, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011. pp. 3201–3208.

[34] A. Gaidon, Z. Harchaoui, C. Schmid, Activity representation with motion hierarchies, Int. J. Comput. Vis. 107 (3) (2014) 219–238.

[35] M. Gales, S. Young, The application of hidden Markov models in speech recognition, Found. Trends Signal Process. 1 (3) (2007) 195–304.

[36] M.A. Goodale, A.D. Milner, 1 2 Separate Visual Pathways for Perception and Action. Essential Sources in the Scientific Study of Consciousness, 2003, 175.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Proc. Advances in Neural Information Processing Systems (NIPS), 2014. pp. 2672–2680.

[38] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, Y. LeCun, Unsupervised learning of spatiotemporally coherent metrics, Proc. Int. Conference on Computer Vision (ICCV), 2015. pp. 4086–4093.

[39] T. Guha, R.K. Ward, Learning sparse representations for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 34 (8) (2012) 1576–1588.

[40] M. Harandi, M. Salzmann, F. Porikli, Bregman divergences for infinite dimensional covariance matrices, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014. pp. 1003–1010.

[41] C. Harris, M. Stephens, A combined corner and edge detector, In Proc. of Fourth Alvey Vision Conference, 1988. pp. 147–151.

[42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[43] M. Hoai, A. Zisserman, Improving human action recognition using score distribution and ranking, Proc. Asian Conference on Computer Vision (ACCV), 2015. pp. 3–20.

[44] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[45] J. Hoffman, E. Rodner, J. Donahue, B. Kulis, K. Saenko, Asymmetric and category invariant feature transformations for domain adaptation, Int. J. Comput. Vis. 109 (1–2) (2014) 28–41.

[46] D. Hogg, Model-based vision: a program to see a walking person, Image Vision Comput. 1 (1983) 5–20.

[47] S. Hongeng, R. Nevatia, Large-scale event detection using semi-hidden Markov models, Proc. Int. Conference on Computer Vision (ICCV), vol. 2, 2003. pp. 1455–1462.

[48] W. Huang, F. Sun, L. Cao, D. Zhao, H. Liu, M. Harandi, Sparse coding and dictionary learning with linear dynamical systems, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[49] T. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, Proc. Advances in Neural Information Processing Systems (NIPS), MIT Press. 1998, pp. 487–493.

[50] M. Jain, H. Jgou, P. Bouthemy, Better exploiting motion for better action recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013. pp. 2555–2562.

[51] H. Jgou, M. Douze, C. Schmid, P. Prez, Aggregating local descriptors into a compact image representation, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2010. pp. 3304–3311.

[52] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 221–231.

[53] Y.G. Jiang, Q. Dai, W. Liu, X. Xue, C.W. Ngo, Human action recognition in unconstrained videos by explicit motion modeling, IEEE Trans. Image Process. 24 (11) (2015) 3781–3795.

[54] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, C.-W. Ngo, Trajectory-based modeling of human actions with motion reference points, Proc. European Conference on Computer Vision (ECCV), 2012. pp. 425–438.

[55] V. Kantorov, I. Laptev, Efficient feature extraction, encoding, and classification for action recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014. pp. 2593–2600.

[56] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014. pp. 1725–1732.

[57] V. Kellokumpu, G. Zhao, M. Pietikinen, Human activity recognition using a dynamic texture based method, British Machine Vision Conference, 2008. pp. 885–894.

[58] A. Kläser, M. MarszaÅek, C. Schmid, A spatio-temporal descriptor based on 3D-gradients, In BMVC08, 2008. pp. 275:1-10.

[59] O. Kliper-Gross, Y. Gurovich, T. Hassner, L. Wolf, Motion interchange patterns for action recognition in unconstrained videos, Proc. European Conference on Computer Vision (ECCV), 2012. pp. 256–269.

[60] P. Koniusz, A. Cherian, F. Porikli, Tensor representations via kernel linearization for action recognition from 3D skeletons, Proc. European Conference on Computer Vision (ECCV), 2016. pp. 37–53.

[61] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space–time neighborhood features for human action recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010. pp. 2046–2053.

[62] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Proc. Advances in Neural Information Processing Systems (NIPS), 2012. pp. 1097–1105.

[63] H. Kuehne, H. Jhuang, R. Stiefelhagen, T. Serre, High Performance Computing in Science and Engineering '12: Transactions of the High Performance Computing Center, Stuttgart (HLRS) 2012, Springer, Berlin Heidelberg, 2013, 571–582. Ch. HMDB51: A Large Video Database for Human Motion Recognition

[64] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, Proc. Int. Conference on Machine Learning (ICML), ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 282–289.

[65] Z. Lan, M. Lin, X. Li, A.G. Hauptmann, B. Raj, Beyond Gaussian pyramid: multi-skip feature stacking for action recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. pp. 204–212.

[66] I. Laptev, On space–time interest points, Int. J. Comput. Vis. 64 (2) (2005) 107–123.

[67] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2008. pp. 1–8.

[68] C. Lea, A. Reiter, R. Vidal, G.D. Hager, Segmental spatiotemporal CNNs for fine–grained action segmentation, Proc. European Conference on Computer Vision (ECCV), 2016. pp. 36–52.

[69] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[70] L.-J. Li, H. Su, L. Fei-fei, E.P. Xing, Object bank: a high-level image representation for scene classification & semantic feature sparsification, Proc. Advances in Neural Information Processing Systems (NIPS), 2010. pp. 1378–1386.

[71] Y. Li, W. Li, V. Mahadevan, N. Vasconcelos, VLAD3: encoding dynamics of deep features for action recognition, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[72] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos "in the wild", Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009. pp. 1996–2003.

[73] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal LSTM with trust gates for 3D human action recognition, Proc. European Conference on Computer Vision (ECCV), Springer International Publishing. 2016, pp. 816–833.

[74] J. Lu, J. Hu, Y.P. Tan, Nonlinear metric learning for visual tracking, 2016 IEEE International Conference on Multimedia and Expo (ICME), 2016. pp. 1–6.

[75] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, IEEE Trans. Image Process. (TIP) 17 (1) (2008) 53–69.

[76] D. Marr, L. Vaina, Representation and recognition of the movements of shapes, Proc. R. Soc. Lond. B Biol. Sci. 214 (1197) (1982) 501–524.

[77] M. Marszalek, I. Laptev, C. Schmid, Actions in context, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009. pp. 2929–2936.

[78] M. Mathieu, C. Couprie, Y. LeCun, Deep Multi-Scale Video Prediction Beyond Mean Square Error, CoRR. 2015.

[79] P. Matikainen, M. Hebert, R. Sukthankar, Trajectons: action recognition through the motion analysis of tracked features, Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, Sept. 2009. pp. 514–521.

[80] A. McCallum, W. Li, Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, CONLL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, 2003, pp. 188–191.

[81] R. Messing, C. Pal, H. Kautz, Activity recognition using the velocity histories of tracked keypoints, Proc. Int. Conference on Computer Vision (ICCV), 2009. pp. 104–111.
[82] D. Metaxas, S. Zhang, A review of motion analysis methods for human non-verbal communication computing, Image Vision Comput. 31 (6–7) (2013) 421–433.
[83] K. Mikolajczyk, H. Uemura, Action recognition with motion-appearance vocabulary forest, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008. pp. 1–8.
[84] I. Misra, C.L. Zitnick, M. Hebert, Unsupervised Learning Using Sequential Verification for Action Recognition, 2016. arXiv preprint arXiv:1603.08561
[85] T.B. Moeslund, E. Granum, A survey of advances in vision-based human motion capture and analysis, Comput. Vis. Image Underst. 104 (3) (2006) 90–127.
[86] J.Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: deep networks for video classification, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. pp. 4694–4702.
[87] B. Ni, X. Yang, S. Gao, Progressively parsing interactional objects for fine grained action detection, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[88] J.C. Niebles, C.-W. Chen, L. Fei-Fei, Modeling temporal structure of decomposable motion segments for activity classification, Proc. European Conference on Computer Vision (ECCV), 2010. pp. 392–405.
[89] E. Norouznezhad, M.T. Harandi, A. Bigdeli, M. Baktash, A. Postula, B.C. Lovell, Directional space–time oriented gradients for 3D visual pattern analysis, Proc. European Conference on Computer Vision (ECCV), 2012. pp. 736–749.
[90] E. Nowak, F. Jurie, B. Triggs, Sampling strategies for bag-of-features image classification, Proc. European Conference on Computer Vision (ECCV), Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 490–503.
[91] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[92] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1? Vis. Res. 37 (23) (1997) 3311–3325.
[93] D. Oneata, J. Verbeek, C. Schmid, Action and event recognition with Fisher vectors on a compact feature set, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013. pp. 1817–1824.
[94] O. Oreifej, Z. Liu, HON4D: histogram of oriented 4D normals for activity recognition from depth sequences, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013. pp. 716–723.
[95] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice, 2014. CoRR abs/1405.4506
[96] X. Peng, C. Zou, Y. Qiao, Q. Peng, Action recognition with stacked Fisher vectors, Proc. European Conference on Computer Vision (ECCV), 2014. pp. 581–595.
[97] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007. pp. 1–8.
[98] R. Poppe, A survey on vision-based human action recognition, Image Vision Comput. 28 (6) (2010) 976–990.
[99] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, T. Darrell, Hidden conditional random fields, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1848–1852.
[100] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (Feb. 1989) 257–286.
[101] H. Rahmani, A. Mian, Learning a non-linear knowledge transfer model for cross-view action recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015. pp. 2458–2466.
[102] H. Rahmani, A. Mian, 3D action recognition from novel viewpoints, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. pp. 1506–1515.
[103] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, S. Chopra, Video (language) Modeling: A Baseline for Generative Models of Natural Videos, CoRR. 2014.
[104] K.K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, Mach. Vis. Appl. 24 (5) (2013) 971–981.
[105] J. Rissanen, Modeling by shortest data description, Automatica 14 (5) (1978) 465–471.
[106] A.J. Robinson, F. Fallside, Static and dynamic error propagation networks with application to speech coding, Proc. Advances in Neural Information Processing Systems (NIPS), 1988. pp. 632–641.
[107] M.D. Rodriguez, J. Ahmed, M. Shah, Action MACH a spatio-temporal maximum average correlation height filter for action recognition, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008. pp. 1–8.
[108] K. Rohr, Towards model-based recognition of human movements in image sequences, CVGIP: Image Underst. 59 (1) (Jan. 1994) 94–115.
[109] M. Rohrbach, S. Amin, M. Andriluka, B. Schiele, A database for fine grained activity detection of cooking activities, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012. pp. 1194–1201.
[110] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[111] S. Sadanand, J.J. Corso, Action bank: a high-level representation of activity in video, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012. pp. 1234–1241.
[112] A. Sanin, C. Sanderson, M.T. Harandi, B.C. Lovell, Spatio-temporal covariance descriptors for action and gesture recognition, IEEE Workshop on Applications of Computer Vision, 2013. pp. 103–110.
[113] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, Proc. Int. Conference on Pattern Recognition (ICPR), ICPR '04, 2004. pp. 32–36.
[114] L. Shao, X. Zhen, D. Tao, X. Li, Spatio-temporal Laplacian pyramid coding for action recognition, IEEE Trans. Cybern. 44 (6) (2014) 817–827.
[115] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, Proc. Advances in Neural Information Processing Systems (NIPS), 2014. pp. 568–576.
[116] B. Singh, T.K. Marks, M. Jones, O. Tuzel, M. Shao, A multi-stream bi-directional recurrent neural network for fine-grained action detection, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[117] G. Somasundaram, A. Cherian, V. Morellas, N. Papanikolopoulos, Action recognition using global spatio-temporal features derived from sparse representations, Comput. Vis. Image Underst. 123 (2014) 1–13.
[118] Y. Song, L.P. Morency, R. Davis, Action recognition by hierarchical sequence summarization, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013. pp. 3562–3569.
[119] K. Soomro, A.R. Zamir, M. Shah, UCF101: A Dataset of 101 Human Actions Classes From Videos in the Wild, 2012. CoRR abs/1212.0402
[120] N. Srivastava, E. Mansimov, R. Salakhutdinov, Unsupervised Learning of Video Representations Using LSTMs, CoRR. 2015.
[121] R.K. Srivastava, K. Greff, J. Schmidhuber, Training very deep networks, Proc. Advances in Neural Information Processing Systems (NIPS), 2015. pp. 2377–2385.
[122] B. Su, J. Zhou, X. Ding, H. Wang, Y. Wu, Hierarchical dynamic parsing and encoding for action recognition, Proc. European Conference on Computer Vision (ECCV), 2016. pp. 202–217.
[123] C. Sun, R. Nevatia, ACTIVE: activity concept transitions in video event classification, Proc. Int. Conference on Computer Vision (ICCV), 2013. pp. 913–920.
[124] C. Sun, R. Nevatia, Large-scale web video event classification by use of Fisher Vectors, Applications of Computer Vision (WACV), 2013 IEEE Workshop on, 2013. pp. 15–22.
[125] L. Sun, K. Jia, D.Y. Yeung, B.E. Shi, Human action recognition using factorized spatio-temporal convolutional networks, Proc. Int. Conference on Computer Vision (ICCV), 2015. pp. 4597–4605.
[126] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, Proc. Advances in Neural Information Processing Systems (NIPS), 2014. pp. 3104–3112.
[127] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. pp. 1–9.
[128] K. Tang, L. Fei-Fei, D. Koller, Learning latent temporal structure for complex event detection, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012. pp. 1250–1257.
[129] Y. Tian, L. Cao, Z. Liu, Z. Zhang, Hierarchical filtered motion for action recognition in crowded videos, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 42 (3) (2012) 313–323.
[130] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3D convolutional networks, Proc. Int. Conference on Computer Vision (ICCV), 2015. pp. 4489–4497.
[131] Y.-H. Hubert Tsai, Y.-R. Yeh, Y.-C. Frank Wang, Learning cross-domain landmarks for heterogeneous domain adaptation, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[132] P. Turaga, R. Chellappa, V.S. Subrahmanian, O. Udrea, Machine recognition of human activities: a survey, IEEE Trans. Circuits Syst. Video Technol. 18 (11) (2008) 1473–1488.
[133] O. Tuzel, F. Porikli, P. Meer, Region covariance: a fast descriptor for detection and classification, Proc. European Conference on Computer Vision (ECCV), 2006. pp. 589–600.
[134] O. Tuzel, F. Porikli, P. Meer, Pedestrian detection via classification on Riemannian manifolds, IEEE Trans. Pattern Anal. Mach. Intell. 30 (10) (2008) 1713–1727.
[135] R.R. Varior, B. Shuai, J. Lu, D. Xu, G. Wang, A Siamese long short-term memory architecture for human re-identification, Proc. European Conference on Computer Vision (ECCV), 2016. pp. 135–153.
[136] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, 2016. arXiv:1604.04494
[137] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3D skeletons as points in a lie group, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014. pp. 588–595.
[138] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, Proc. Int. Conference on Machine Learning (ICML), 2008. pp. 1096–1103.
[139] S. Vishwakarma, A. Agrawal, A survey on activity recognition and behavior understanding in video surveillance, Vis. Comput. 29 (10) (2013) 983–1009.
[140] H. Wang, A. Klaser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011. pp. 3169–3176.
[141] H. Wang, C. Schmid, Action recognition with improved trajectories, Proc. Int. Conference on Computer Vision (ICCV), 2013. pp. 3551–3558.
[142] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, British Machine Vision Conference, Sep. 2009. pp. 127.

[143] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. pp. 4305–4314.
[144] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, Towards Good Practices for Very Deep Two-Stream ConvNets, 2015. CoRR abs/1507.02159
[145] X. Wang, A. Farhadi, A. Gupta, Actions ~ transformations, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. pp. 2658–2667.
[146] X. Wang, A. Gupta, Unsupervised learning of visual representations using videos, Proc. Int. Conference on Computer Vision (ICCV), 2015. pp. 2794–2802.
[147] Y. Wang, G. Mori, Hidden part models for human action recognition: probabilistic versus max margin, IEEE Trans. Pattern Anal. Mach. Intell. 33 (7) (2011) 1310–1323.
[148] D. Weinland, R. Ronfard, E. Boyer, Free viewpoint action recognition using motion history volumes, Comput. Vis. Image Underst. 104 (23) (2006) 249–257.
[149] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Comput. Vis. Image Underst. 115 (2) (2011) 224–241.
[150] G. Willems, T. Tuytelaars, L. Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, Proc. European Conference on Computer Vision (ECCV), 2008. pp. 650–663.
[151] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[152] J. Wu, Y. Zhang, W. Lin, Towards good practices for action video encoding, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014. pp. 2577–2584.
[153] Z. Wu, Y. Jiang, X. Wang, H. Ye, X. Xue, J. Wang, Fusing Multi-Stream Deep Networks for Video Classification, CoRR. 2015.
[154] D. Xing, X. Wang, H. Lu, Action recognition using hybrid feature descriptor and VLAD video encoding, Computer Vision — ACCV 2014 Workshops: Singapore, Singapore, November 1–2, 2014, Revised Selected Papers, Part I, 2015. pp. 99–112.
[155] X. Yan, H. Chang, S. Shan, X. Chen, Modeling video dynamics with deep dynencoder, Proc. European Conference on Computer Vision (ECCV), 2014. pp. 215–230.
[156] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. 38 (4). (2006)
[157] A. Yilmaz, M. Shah, Actions sketch: a novel action representation, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, 2005. pp. 984–989.
[158] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, Proc. European Conference on Computer Vision (ECCV), 2014. pp. 818–833.
[159] B. Zhan, D.N. Monekosso, P. Remagnino, S.A. Velastin, L.-Q. Xu, Crowd analysis: a survey, Mach. Vis. Appl. 19 (5) (2008) 345–357.
[160] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 915–928.
[161] Y. Zhu, X. Zhao, Y. Fu, Y. Liu, Sparse coding on local spatial–temporal volumes for human action recognition, Proc. Asian Conference on Computer Vision (ACCV), Springer. 2011, pp. 660–671.