

Video Imprint

Zhanning Gao, Le Wang, *Member, IEEE*, Nebojsa Jojic, Zhenxing Niu, *Member, IEEE*, Nanning Zheng, *Fellow, IEEE*, Gang Hua, *Senior Member, IEEE*

Abstract—A new unified video analytics framework (ER3) is proposed for complex event retrieval, recognition and recounting, based on the proposed video imprint representation, which exploits temporal correlations among image features across video frames. With the video imprint representation, it is convenient to reverse map back to both temporal and spatial locations in video frames, allowing for both key frame identification and key areas localization within each frame. In the proposed framework, a dedicated feature alignment module is incorporated for redundancy removal across frames to produce the tensor representation, *i.e.*, the video imprint. Subsequently, the video imprint is individually fed into both a reasoning network and a feature aggregation module, for event recognition/recounting and event retrieval tasks, respectively. Thanks to its attention mechanism inspired by the memory networks used in language modeling, the proposed reasoning network is capable of simultaneous event category recognition and localization of the key pieces of evidence for event recounting. In addition, the latent structure in our reasoning network highlights the areas of the video imprint, which can be directly used for event recounting. With the event retrieval task, the compact video representation aggregated from the video imprint contributes to better retrieval results than existing state-of-the-art methods.

Index Terms—Event videos, Feature alignment, Feature aggregation, Reasoning network.

1 INTRODUCTION

ANALYSIS of event videos is generally considered more challenging than the related video-based action recognition task [6], [41], thanks to the richer contents of such event videos. Typical event videos are much longer (several minutes or even hours) than trimmed action recognition videos, and multiple human actions and a variety of different objects often appear across various scenes. For example, a “birthday party” event may take place at home or in a restaurant, with multiple objects coming into focus, *e.g.*, a birthday cake, and may include a variety of activities that span multiple frames, *e.g.*, singing the birthday song, or blowing out candles.

In the last decade, analysis of complex events in videos has attracted significant attention in the computer vision community [5], [10], [13], [14], [22], [31], [38], [44], [49]. Previous research could be categorized as the unsupervised and the supervised methods. Unsupervised methods were typically used for *event retrieval* [12], [38] where the goal is the retrieval of all related videos semantically relevant to the query video sample. On the other hand, supervised learning has been widely used in *event recognition* [4], [7] and detection tasks [31], [57] in similar ways to its applications in action recognition [6], [41], [61] and generic video classification tasks [23], [56], [59]. In the latter case, a classifier is trained on the annotated training set to recognize the event

- Z. Gao, L. Wang and N. Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China. E-mail: zhanninggao@gmail.com, {lewang, nnzheng}@mail.xjtu.edu.cn.
- N. Jojic and G. Hua are with the Microsoft Research, Redmond, Washington, WA 98052. E-mail: jojic@microsoft.com, ganghua@gmail.com.
- Z. Niu is with Alibaba Group, Hangzhou, Zhejiang 311121. E-mail: zhenxing.nzx@alibaba-inc.com.

Manuscript received 16 Dec. 2017; revised 10 July 2018; accepted 12 Aug. 2018. Date of publication XX XXX 2018; date of current version XX XXX 2018.

(Corresponding author: Le Wang.)

Recommended for acceptance by XXX.

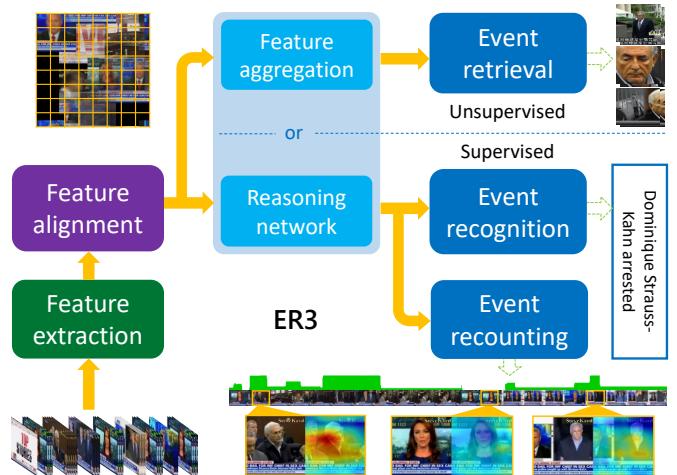


Fig. 1. Illustration of the ER3 framework for event retrieval, recognition and recounting. The compact video representation from feature aggregation can be used for large-scale event retrieval. With supervised training, ER3 can also recognize the event category of the input video. Event recounting falls directly out of the latent structure of the model in form of statistics displayed as heat maps for each frame indicating key areas related to the event.

categories of test videos, *e.g.*, the multimedia event detection task of the TRECVID [33]. In practical applications, it is often desirable to provide explainable results by qualifying the category prediction with the localization of the key pieces of evidence that lead to the recognition decision, which is sometimes referred to as the *event recounting*.

A major challenge in *event retrieval* and *event recognition* is the construction of appropriate video representations, which should ideally be both discriminative for efficient disambiguation and compact for computational efficiency. Conventionally, a video representation is a fixed-length per-video global feature vector extracted from many frame-

level appearance features [12], [13], [38], [56], [57]. In a typical *event recognition* task, this video representation is fed into a linear classifier [57] or a neural network [23], [56] for classification. However, this procedure is generally incompatible with event recounting, as tracing the decision back to individual frame locations is impossible due to the irreversible video representation (global feature) extraction. Therefore, most existing methods perform event recounting as an extra post-processing step [13], [29], [50].

To address these challenges, the ER3 framework is proposed to simultaneously achieve *event retrieval*, *recognition* and *recounting*. Figure 1 illustrates the components and the inputs/outputs of such ER3 system. In ER3, (i) we introduce a feature alignment step which can significantly suppress the redundant information and generate a more comprehensive and compact video representation called *video imprint*. In addition, the video imprint also preserves the local spatial layout among video frames. (ii) Based on the video imprint, in unsupervised setting, we propose an efficient aggregation method for large-scale event retrieval. In supervised setting, we further employ a reasoning network, a modified version of the neural memory networks [43], which can simultaneously recognize the event category and locate the key pieces of evidence for the event category. In fact, the recounting is so naturally integrated in the framework that the experiments show that the recounting step can assist the recognition task and improve the recognition accuracy. (iii) In the recounting task, both temporal key frame identification (attribution of the key frames with respect to the event category, as in [29], [50]) and spatial key areas localization (attribution of the key areas within each frame) are implemented, thanks to the video imprint preserving local semantic and spatial layouts.

This manuscript is an extension of our conference paper [15] with modifications as follows. In the feature alignment step, an alternative generative model (*i.e.*, epitome [25]) is included besides the the tessellated counting grid (TCG) model [34], [35]. Both generative models share the core idea of building condensed representation by exploiting the spatial interdependence among the input features. However, unlike the TCG model which is limited to counts/histogram-style input features, the epitome model is represented as structured Gaussian mixtures which can accommodate general vector or tensor input features. In addition, to accelerate the feature alignment step with the epitome model, we propose an accelerated two-step scheme to update the epitome. More details are provided in Section 3.2. We also provide a comprehensive comparison between the two generative models with various datasets and tasks. The experimental results show that the alternative epitome model achieves higher computational efficiency with comparable results with the TCG model.

The paper is organized as follows. Section 2 discusses related work about event videos analysis. Then, we present the technical details of the ER3 in Section 3. Experimental results are provided in Section 4. Finally, we conclude the paper in Section 5.

2 RELATED WORK

With a typical unsupervised event retrieval task, the goal is the retrieval of all related videos semantically relevant to the query video sample. A major challenge is the construction of both compact and discriminative video representations. Conventional methods [12], [37], [38] rely on frame-level local features (*e.g.*, SIFT [30]) and aggregation strategies (*e.g.*, Fisher Vector [36], [39], VLAD [12], [20], [21] or explicit feature maps [37]) for frame-level feature description. Recent works [23], [57] predominantly employ deep Convolutional Neural Network (CNNs) [18], [42] to extract a feature descriptor from each video frame. Latest work [3] revisits temporal match kernels [37] and presents a learnable temporal layer which can further enhance the CNNs based feature descriptor. Subsequently, a video-level representation is typically obtained by directly averaging all frame-level descriptors in the video. Such sum-aggregation strategy disregards the strong temporal correlations among consecutive frames, which may over-emphasize certain long or recurrent shots in the video. We discuss this problem in Section 3 and show that the redundant information among frames can be effectively suppressed by the feature alignment step.

Event recognition and detection have attracted wide attention in the last decade. In general, event video recognition system usually consists of three stages, *i.e.*, feature extraction, feature aggregation/pooling and training/recognition. As in event retrieval, the first two stages aim at building discriminative video representations. Previous methods focused mostly on designing better video features or representations for the classifier, such as hand-crafted visual features [11], [30], motion features [51], [52], audio features [2], and mid-level concept/attribute features [9], [48]. Recently, the advancements of CNN [27], [42] lead to promising results in event recognition task [23], [57], [60]. The video representations are usually constructed by direct aggregation of the frame-level CNN features. Due to limited amount of training data, video representations are typically fed to classifiers such as the Support Vector Machine (SVM) [8] or multi-feature fusion framework [23], [56], [60], [62].

Event recounting refers to the attribution of key pieces of evidence supporting the recognition decision. As with most video analytics datasets, only video-level annotations are provided, making such attribution a challenging task. Event recounting is usually implemented as a post-processing step after the recognition [29], [50]. Chang *et al.* [9] proposed a joint optimization framework with mid-level semantic concept representations for event recognition and recounting. Sun *et al.* [45] introduced an evidence localization model learned via a max-margin framework, and Lai *et al.* [28] applied Multiple-Instance Learning (MIL) which infers temporal instance labels and the video-level labels. In these works, event videos are treated as sets of shots or instances, these recounting procedures perform only temporal localization and usually at a coarse scale. Recently, Gan *et al.* [13] proposed a deep neural network for event recognition. Specifically, event recounting (both temporal and spatial) is also achieved via passing the classification scores backward. However, this is still an explanatory post-processing step which is never designed to assist the event recognition task.

In contrast with these methods, at the core of our integrated event recognition and recounting framework is a la-

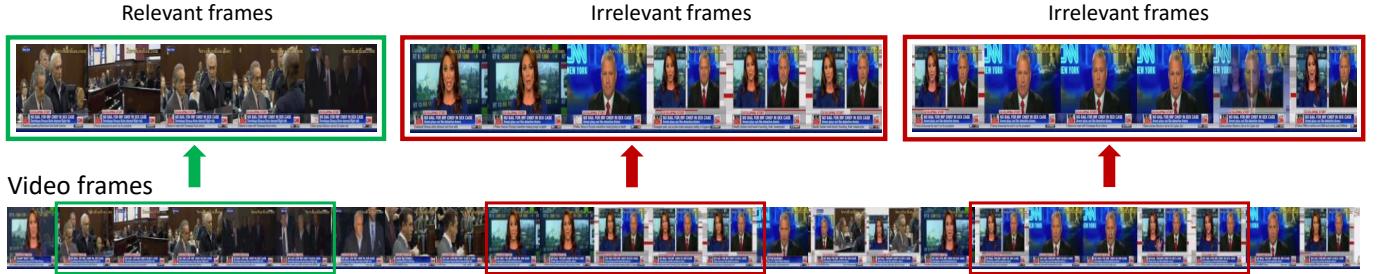


Fig. 2. Illustration of the frames related to “Dominique Strauss-Kahn arrested”. The frames in green box denote the positive frames related to the event. Red boxes show irrelevant frames.

tent structure that contains reverse attribution pointers back to the video frames. In addition, the proposed framework is trained in an unsupervised manner by simultaneously aligning areas across frames and estimating a probability distribution over frame features in the corresponding areas. The obtained representation consists of a grid of distributions with the corresponding mappings from the feature to the frames, similar to the way video frames are mapped to panoramas from pixel space (see the toy examples in Figure 1 and Figure 3). The obtained grid and such reverse mapping pointers (back to the spatial locations of the specific video frames) form the proposed *video imprint*. With this video imprint representation, it is possible to design an aggregation to emphasize mere presence instead of frequency of repetition.

Our experiments show that the proposed video imprint aggregation yields better performances in both the supervised and unsupervised tasks than existing algorithms. The video imprint also allows for the reasoning over the spatial layouts of features across frames. Inspired by the attention mechanism in memory networks [43] reasoning over sentences in priming text and video face recognition [58], the proposed reasoning network analyzes evidences at different spatial locations of the compact video imprint while carrying out the recognition task, which also highlights the key areas of the imprint. These key areas of the imprint could readily be mapped back to specific video frames and corresponding spatial locations. In this way, event recounting is implemented as an integral part of recognition instead of a post-processing step.

3 THE DETAILS OF ER3

In this section, we present all modules and the operating mechanism of the proposed ER3 framework, as illustrated in Figure 1.

3.1 Feature extraction

Recently, image descriptors based on the activations of convolutional layers [17], [40], [60] have outperformed previous methods [23], [60] based on features extracted from fully connected layers. Inspired by the spatial information preserving characteristics of convolutional layers outputs, we also choose the activations of the last convolutional layer as the frame-level feature.

3.2 Feature alignment

Existing event recognition algorithms rely on a compact and discriminative video representation, which is typically direct average of the frame-level descriptors [12], [23], [57], [60]. Such sum-aggregation strategy disregards the strong temporal correlations among consecutive frames, which may over-emphasize certain long or recurrent shots in the video. Therefore, irrelevant and repetitive shots in the video might dominate the obtained video representation. For instance, in the event “Dominique Strauss-Kahn arrested” as shown in Figure 2, many video frames showing the news anchor are irrelevant, but they are visually similar, therefore the simple averaging aggregation strategy for video representation may over-emphasize such irrelevant frames. To mitigate this problem, we propose the feature alignment procedure to regularize the influence of frame features.

The idea of feature alignment is inspired by panoramic stitching [46], [47], which can remove the redundant or overlapping parts between multiple images. The redundancy across video frames could likewise be significantly reduced to improve the robustness of the obtained video representation against long and/or repetitive irrelevant video frames.

Due to the dynamic and complex nature of event videos, it is impractical to directly stitch video frames by pixel-level alignment. Instead, image features are extracted first using the activations of the last convolutional layer with each frame as input. Afterwards, the tessellated counting grid (TCG) [34], [35] or epitome model [25] is employed to generate a tensor of frame-level feature distributions, which implies a panorama-style reversible mapping, accommodating the geometric variations in objects and scenes in event videos. This model exploits the spatial interdependence of the frame-level features in relevant frames, which makes possible the subsequent clustering of visually similar shots.

The inquisition for leveraging the epitome model [25] is to accelerate the feature alignment step. Both the TCG and the epitome can capture the spatial interdependence among input features. However, the input features of the epitome model can be more flexible, considering their Gaussian location distribution instead of the discrete categorical distribution in the TCG. In addition, the incorporation of the epitome model also allows us to propose a highly efficient two-step implementation, which leads to approximately an order of magnitude faster feature alignment procedure than the counterpart in the TCG. In the remainder of this section,

TABLE 1
Principal Notations.

π_i	Parameters of the counting grid model. l_1 —normalized counts feature at location i on the counting grid $E = [1 \dots E_x] \times [1 \dots E_y]$
$\pi_{i,z}$	z -th dimension of π_i and $\sum_{z \in Z} \pi_{i,z} = 1$, where $Z \in \mathbb{Z} = [1, \dots, Z]$
(μ_i, σ_i^2)	Parameters of the epitome model. The mean μ_i and variance σ_i^2 of the Gaussian distribution aligned at the location i of the grid E
$\{c^s\}_{s \in S}$	Counts features plugged in a tessellation S , where $s \in S = [1 \dots S_x] \times [1 \dots S_y]$
F	Tensor features, e.g., the activations from convolutional layer of the CNN model
f_j	The feature vector extracted from F along the spatial dimensions, i.e., $j \in W = [1, \dots, W_x] \times [1, \dots, W_y]$
W_k	The window at the location k of the grid E which assumed to generate counts features $\{c^s\}_{s \in S}$ or tensor features F
W_k^s	The sub-window in the W_k generating each c^s
l	The latent variable that represents the mapping location in the grid E

mathematical notations are first summarized in Table 1, followed by introductions of both generative models¹.

3.2.1 Tessellated counting grid

Tessellated counting grid (TCG) [34], [35] is designed to capture the spatial interdependence among image features. Given a set of images or a video sequence, it assumes that each image/frame is represented by a set of l_1 —normalized non-negative feature vectors (e.g., bags of visual words vectors) $\{c^s\}_{s \in S}$ plugged in a tessellation $S = [1, \dots, S_x] \times [1, \dots, S_y]$ ².

Formally, the counting grid π_i is a set of normalized counts of features indexed by $z \in Z = [1 \dots Z]$ (dimension of image feature) on the 2D discrete grid $i = (i_x, i_y) \in E = [1, \dots, E_x] \times [1, \dots, E_y]$, where i denotes the location on the grid and $\sum_{z \in Z} \pi_{i,z} = 1$ [26].

As a generative model, the image features $\{c^s\}_{s \in S}$ are assumed to follow a distribution found in a window into the counting grid. The probability of generating the image features $\{c^s\}_{s \in S}$ from the window $W_k = [k_x, \dots, k_x + W_x - 1] \times [k_y, \dots, k_y + W_y - 1]$ placed at the location $k = (k_x, k_y) \in E$ of the grid is

$$p(\{c^s\}_{s \in S} | l = k) = \gamma \prod_{z \in Z} \prod_{s \in S} \left(\sum_{i \in W_k^s} \pi_{i,z} \right)^{c_z^s}, \quad (1)$$

where γ is the normalization constant. l denotes the latent variable, while i and k represent generic positions in the grid E . Then, for a given counting grid π , the joint distribution over the set of image features $\{c^{s,t}\}_{s \in S, t \in T}$, indexed by

1. Comprehensive introductions to both the TCG and the epitome models can be find in [25], [34], [35].

2. With l_1 —normalization and appropriate down-sampling, the feature maps (after ReLU) from the convolutional layer of CNN model naturally satisfy this assumption.

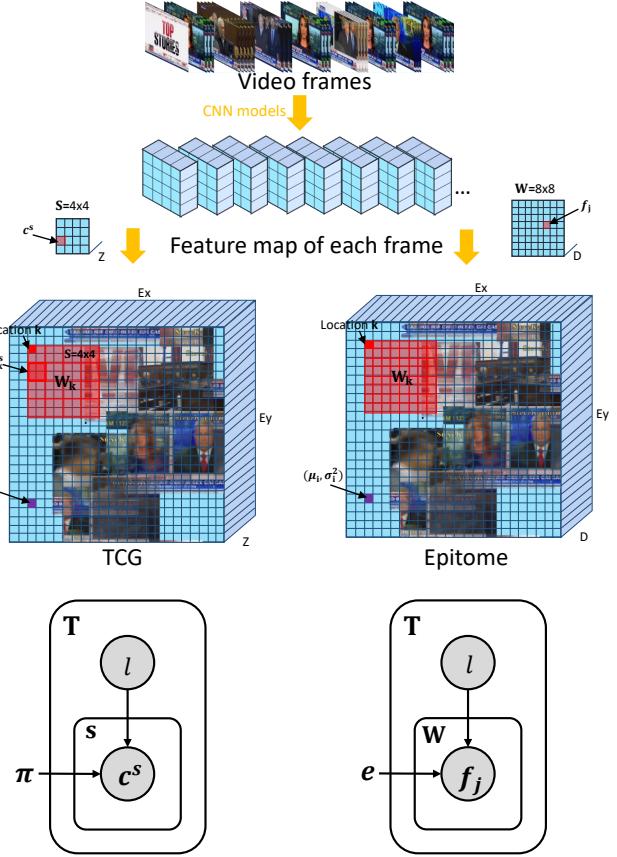


Fig. 3. Illustration of tessellated counting grid (TCG) and epitome models, and their Bayesian networks. The left tensor block represents the TCG with $E = 24 \times 24$, $W = 8 \times 8$, $S = 4 \times 4$. The right tensor block represents the epitome with $E = 24 \times 24$, $W = 8 \times 8$. For TCG, the input CNN feature maps are down-sampled to $S = 4 \times 4$. In fact, the epitome can be regarded as a special case of TCG when $W = S$. For both TCG and epitome, similar frames are usually represented in the same or nearby windows, e.g., the anchor who we frequently see in the video.

$t \in T = [1 \dots T]$, and their corresponding latent window locations $\{l^t\}$ in the grid can be derived as

$$P(\{c^{s,t}\}_{s \in S, t \in T}, \{l^t\}_{t \in T}) \propto \prod_{t \in T} \sum_{k \in E} \prod_{z \in Z} \prod_{s \in S} \left(\sum_{i \in W_k^s} \pi_{i,z} \right)^{c_z^{s,t}}. \quad (2)$$

The counting grid π can be estimated by maximizing the log likelihood of the joint distribution with an EM algorithm,

$$\text{E step : } q(l^t = k) \propto \exp \left(\sum_{s \in S} \sum_{z \in Z} c_z^{s,t} \log \sum_{i \in W_k^s} \pi_{i,z} \right), \quad (3)$$

$$\text{M step : } \pi_{i,z} \propto \pi_{i,z}^{old} \sum_{t \in T} \sum_{s \in S} c_z^{s,t} \sum_{k|i \in W_k^s} \frac{q(l^t = k)}{\sum_{i \in W_k^s} \pi_{i,z}^{old}},$$

where $q(l^t = k)$ denotes the posterior probability $p(l^t = k | \{c^{s,t}\}_{s \in S})$ and $\pi_{i,z}^{old}$ denotes the counting grid at the previous iteration.

The iterative process of TCG jointly estimates the counting grid (*i.e.*, *video imprint*) π and aligns all video frame features to it with such correspondences captured in q .

3.2.2 Epitome

The original epitome model [25] takes raw pixels as input and aims to mine the essence of the textural and shape properties of the image. Formally, the epitome \mathbf{e} is a set of dependent Gaussian distributions $\{\mathcal{N}(\mathbf{f}_j; \boldsymbol{\mu}_i, \sigma_i^2)\}$ aligned on a grid $\mathbf{i} = (i_x, i_y) \in \mathbf{E} = [1, \dots, E_x] \times [1, \dots, E_y]$ just like TCG. In the original epitome formulation, f_j denotes the intensity or the color of the pixel on the image patch \mathbf{F} indexed by $\mathbf{j} = (j_x, j_y) \in \mathbf{W} = [1, \dots, W_x] \times [1, \dots, W_y]$. Here we extend \mathbf{F} to a general tensor (feature map), specifically, the activations from the last convolutional layer of a CNN model. In other words, \mathbf{F} is the CNN feature map and f_j is the feature vector extracted from \mathbf{F} along the spatial dimensions, *i.e.*, $\mathbf{j} \in \mathbf{W}$.

Given the epitome \mathbf{e} , the probability of generating the feature map \mathbf{F} from the window \mathbf{W}_k at location k of the epitome \mathbf{e} is

$$p(\mathbf{F}|l = k) = \gamma \prod_{i \in \mathbf{W}_k} \mathcal{N}(f_{i-k}; \boldsymbol{\mu}_i, \sigma_i^2), \quad (4)$$

where γ is the normalization constant. $\mathcal{N}(f_{i-k}; \boldsymbol{\mu}_i, \sigma_i^2)$ is a Gaussian distribution over f_{i-k} with mean $\boldsymbol{\mu}_i$ and variance σ_i^2 and $\mathbf{i} - \mathbf{k} = (i_x - k_x + 1, i_y - k_y + 1)$. Similar to TCG, the joint distribution over the set of feature maps $\{\mathbf{F}^t\}_{t \in \mathbf{T}}$, indexed by t , and their corresponding latent window locations $\{l^t\}_{t \in \mathbf{T}}$ on the epitome can be derived as

$$P(\{\mathbf{F}^t\}, \{l^t\}) \propto \prod_{t \in \mathbf{T}} \sum_{k \in \mathbf{E}} \prod_{i \in \mathbf{W}_k} \mathcal{N}(f_{i-k}^t; \boldsymbol{\mu}_i, \sigma_i^2). \quad (5)$$

The parameters $(\boldsymbol{\mu}, \sigma^2)$ are estimated by marginalizing the joint distribution, *i.e.*, optimizing the log likelihood of the data with an iterative EM algorithm,

$$\begin{aligned} \text{E step : } q(l^t = k) &\propto \prod_{i \in \mathbf{W}_k} \mathcal{N}(f_{i-k}^t; \boldsymbol{\mu}_i, \sigma_i^2), \\ \text{M step : } \boldsymbol{\mu}_i &= \frac{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} q(l^t = k) f_{i-k}^t}{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} q(l^t = k)}, \\ \sigma_i^2 &= \frac{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} q(l^t = k) (f_{i-k}^t - \boldsymbol{\mu}_i)^2}{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} q(l^t = k)}, \end{aligned} \quad (6)$$

where $q(l^t = k)$ denotes the posterior probability $p(l^t = k | \mathbf{F}^t)$ and $\mathbf{i} - \mathbf{W} = (i_x - W_x + 1, i_y - W_y + 1)$.

As presented above, both generative models (TCG and epitome) build condensed representations by capturing the spatial interdependence among input features. The epitome model differs from the TCG in the location distributions (Gaussian versus discrete categorical). Therefore, the input features of the epitome can be more flexible. See Figure 3 for the illustration of the TCG and epitome³. In addition, according to Equation (6), the updating of the epitome parameters, *i.e.*, the M step, only depends on the current q distribution, *i.e.*, the current E step, and the input features. This motivates us to propose an efficient two-step scheme to generate the epitome based video imprint, *i.e.*, we can first efficiently estimate the final q distribution, and then calculate the epitome based video imprint directly with

³ For ease of illustration of the video imprint, we accumulate the frames on the location with the maximum posterior probability $q(l^t = k)$ and draw the mean image.

Algorithm 1 The efficient two-step scheme

Input: the feature maps $\{\mathbf{F}^t\}_{t \in \mathbf{T}}$, epitome size \mathbf{E} , window size \mathbf{W}
Output: $\hat{q}, \boldsymbol{\mu}$

- 1: **for** each $t = 1 \dots T$ **do**
- 2: $\mathbf{G}^t = \phi_{PCA}(\mathbf{F}^t)$
- 3: **end for**
- 4: **repeat**
- 5: Update \hat{q} with the EM steps of Equation (7)
- 6: **until** Convergence
- 7: Compute $\boldsymbol{\mu}$ with Equation (8)
- 8: Return $\hat{q}, \boldsymbol{\mu}$

Equation (6). The following are the details of the efficient two-step implementation.

3.2.3 An efficient two-step scheme

Given a fixed epitome size \mathbf{E} and window size \mathbf{W} , the only feasible way to accelerate the learning procedure of epitome model is to reduce the dimension of the input feature maps $\{\mathbf{F}^t\}_{t \in \mathbf{T}}$. Although the low dimensional input features produced from $\{\mathbf{F}^t\}_{t \in \mathbf{T}}$ by dimension reduction are enough to capture the correspondences among frames within a video, it may also reduce the discriminative capacity of the epitome based video imprint $\boldsymbol{\mu}$ for large-scale video event analysis. Therefore, we propose an efficient two-step scheme which divides the epitome based feature alignment operation into two steps: the correspondence analysis step and the video imprint generation step.

At the first step, the low dimensional feature maps $\{\mathbf{G}^t\}_{t \in \mathbf{T}}$ are generated by PCA [19] projection, $\mathbf{G} = \phi_{PCA}(\mathbf{F})$, where $\mathbf{G} \in \mathbb{R}^{W_x \times W_y \times d}$, $\mathbf{F} \in \mathbb{R}^{W_x \times W_y \times D}$ and $d \ll D$. Then, the epitome of $\{\mathbf{G}^t\}_{t \in \mathbf{T}}$ can be learned with

$$\begin{aligned} \text{E step : } \hat{q}(l^t = k) &\propto \prod_{i \in \mathbf{W}_k} \mathcal{N}(g_{i-k}^t; \hat{\boldsymbol{\mu}}_i, \hat{\sigma}_i^2), \\ \text{M step : } \hat{\boldsymbol{\mu}}_i &= \frac{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} \hat{q}(l^t = k) g_{i-k}^t}{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} \hat{q}(l^t = k)}, \\ \hat{\sigma}_i^2 &= \frac{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} \hat{q}(l^t = k) (g_{i-k}^t - \hat{\boldsymbol{\mu}}_i)^2}{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} \hat{q}(l^t = k)}. \end{aligned} \quad (7)$$

As an approximation of the q distribution, we found that \hat{q} can fully capture the correspondence of video frames (see Figure 5). In addition, since $d \ll D$, \hat{q} estimation with Equation (7) achieves much higher efficiency.

At the second step, according to the M step of the Equation (6), the updated parameters $\boldsymbol{\mu}$ and σ^2 only depend on the set of input feature maps $\{\mathbf{F}^t\}_{t \in \mathbf{T}}$ and the q distribution from the last iteration of E step. Therefore, if we replace q with its approximation \hat{q} from the first step, $\boldsymbol{\mu}$ and σ^2 can be directly computed with

$$\begin{aligned} \boldsymbol{\mu}_i &= \frac{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} \hat{q}(l^t = k) f_{i-k}^t}{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} \hat{q}(l^t = k)}, \\ \sigma_i^2 &= \frac{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} \hat{q}(l^t = k) (f_{i-k}^t - \boldsymbol{\mu}_i)^2}{\sum_{t \in \mathbf{T}} \sum_{k \in \mathbf{W}_{i-W}} \hat{q}(l^t = k)}. \end{aligned} \quad (8)$$

The computational complexity and performance of the efficient two-step scheme will be discussed in Section 4. We

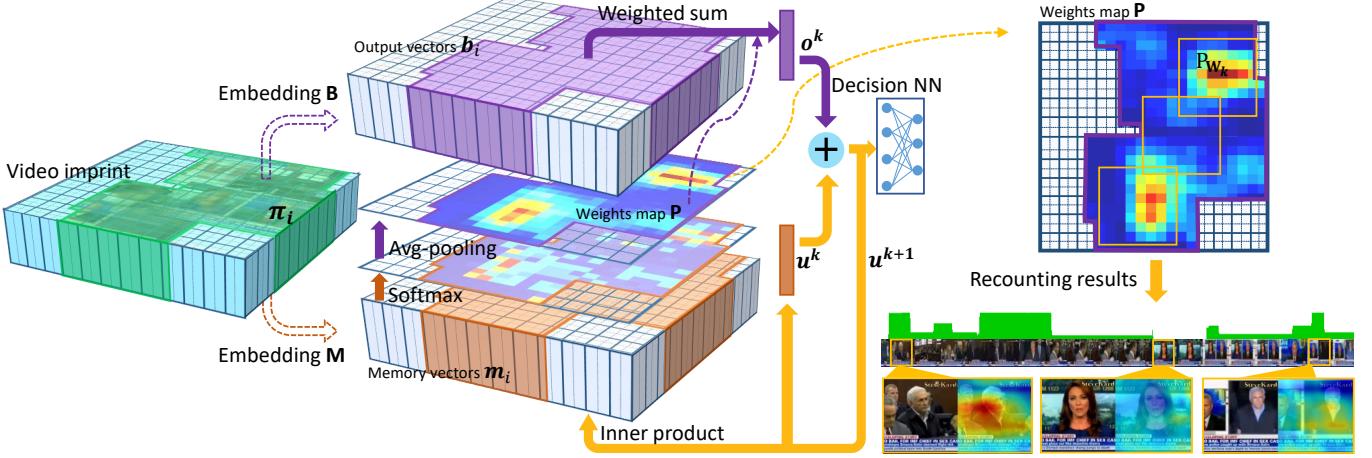


Fig. 4. Illustration of the reasoning network for event recognition and event recounting.

shall note that this fast implementation scheme is inapplicable to TCG. It is difficult to transfer a high dimensional histogram vector to a low dimensional space. Most importantly, as shown in Equation (3), the current π_i relies on both π_i^{old} and the q distribution at the previous iteration which makes the efficient two-step scheme invalid for TCG. In practice, $\hat{\sigma}_i^2$ is fixed as 0.1 to avoid overfitting during the EM iterations and only μ is taken as video imprint which has the same size with π . The efficient two-step scheme is summarized in Algorithm 1.

3.3 Feature aggregation

In this section, we demonstrate how to aggregate the video imprint into a compact video representation for unsupervised event retrieval. We refer each π_i or μ_i on the video imprint as an imprint descriptor. As shown in Figure 3, some imprint descriptors are meaningless since no frames are aligned to their locations. The first step is to generate an active map for the video imprint to eliminate these meaningless imprint descriptors. Formally, the binary active map, $\mathbf{A} = \{a_i | i \in \mathbf{E}\}$, $a_i \in \{0, 1\}$, is computed as

$$a_i = \begin{cases} 1 & \left\{ i \in \mathbf{W}_k | k : \sum_{t=1}^T q(l_t = k) > \tau \right\}, \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where τ is the threshold of the active map.

After the active map is obtained, a direct sum-aggregation over the entire activated imprint descriptors is carried out to produce the final video representation. Formally, the aggregation step can be written as

$$\begin{aligned} \text{TCG} : \phi_{FA}(\pi, \mathbf{A}) &= \sum_{i \in \mathbf{E}} a_i \pi_i, \\ \text{Epitome} : \phi_{FA}(\mu, \mathbf{A}) &= \sum_{i \in \mathbf{E}} a_i \mu_i. \end{aligned} \quad (10)$$

The obtained ϕ_{FA} is subsequently l_2 -normalized and the cosine similarity is computed for event retrieval.

3.4 Reasoning over the imprint

Once the imprint is obtained for each video, it makes reasoning based on this compact imprint representation possible, where each location corresponds to a recurring scene/object part, with spatial layouts of imprint locations reflecting the scene/object spatial layouts in video frames where they were observed. We treat locations in the imprint in a similar way to the sentences in memory networks [43], [53]. Our reasoning network determines the event category in stages, which traverses attention from one set of imprint locations to the next. In this process as illustrated in Figure 4, the imprint locations of large importance are highlighted, and we also trace these highlights back to the locations in video frames using the q/\hat{q} distributions as discussed above.

Our reasoning network differs from the memory networks in two ways. First, since there is no query question for event recognition, we initialize the input vector u^1 with Equation (10), *i.e.*, sum-aggregation of video imprint. Second, because the spatial organization in the imprint is meaningful, we incorporate an average spatial pooling layer after the softmax layer to improve the smoothness of the recounting results. The model details are as follows.

Memory layers in the reasoning network. As shown in Figure 4, the video imprint (non-activated locations are ignored) is processed via multiple memory layers (hops). In each layer, the imprint descriptors π_i or μ_i from video imprint are first embedded to the output vector space and memory vector space with embedding matrices \mathbf{B} and \mathbf{M} , respectively,

$$\begin{aligned} \text{TCG} : \mathbf{b}_i &= \mathbf{B}\pi_i, \quad \mathbf{m}_i = \mathbf{M}\pi_i, \\ \text{Epitome} : \mathbf{b}_i &= \mathbf{B}\mu_i, \quad \mathbf{m}_i = \mathbf{M}\mu_i, \end{aligned} \quad (11)$$

where \mathbf{b}_i denotes the output vector and \mathbf{m}_i denotes memory vector. The memory vector \mathbf{m}_i is introduced to compute the weights map $\mathbf{P} = \{p_i | i \in \mathbf{E}\}$ with the internal state \mathbf{u} ,

$$p_i = \text{avgpooling}(\text{softmax}(\mathbf{u}^\top \mathbf{m}_i)). \quad (12)$$

The average pooling is performed with 3×3 windows, stride 1. The output vector \mathbf{o} is then computed by a weighted sum over the output vectors \mathbf{b}_i , *i.e.*,

$$\mathbf{o} = \sum_i p_i \mathbf{b}_i. \quad (13)$$

For the internal state vector \mathbf{u} , the initial \mathbf{u}^1 is obtained with Equation (10), and the \mathbf{u}^{k+1} in $k + 1$ layer can be computed by

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{o}^k. \quad (14)$$

The obtained output vector is fed into a decision network for event categorization. The decision network can consist of only a single softmax layer or multiple fully connected layers. The recounting heat map⁴ (posterior probabilities $q(l^t = \mathbf{i})$) of each frame shown in Figure 4 is generated via the sum of all weights maps, $\mathbf{P}^{\text{sum}} = \sum_k \mathbf{P}^k$. We use $\mathbf{P}_{\mathbf{W}_i}^{\text{sum}}$ to denote the weights map cropped from \mathbf{P}^{sum} in the window \mathbf{W}_i . Then the recounting map \mathbf{R}^t of frame t is

$$\mathbf{R}^t = \sum_{i \in \mathbf{E}} q(l^t = \mathbf{i}) \mathbf{P}_{\mathbf{W}_i}^{\text{sum}}. \quad (15)$$

The importance score of each frame is obtained with the sum of the recounting map.

4 EXPERIMENTS

4.1 Datasets and evaluation protocol

In terms of event retrieval, we validated our method on the large-scale benchmark EVVE dataset [38]. It contains 2,995 videos (620 videos are set as queries) related to 13 specific event classes. Given a single video of an event, the task is to retrieve videos related to the same event from the dataset. The methods are evaluated based on the mean AP (mAP) computed per event. The overall performance is evaluated by averaging the mAPs over the 13 events. In addition, a large distractor dataset (100,000 vedios) is also provided to evaluate the retrieval performance on large-scale data.

To evaluate the event recognition and recounting, we used three datasets: EVVE, Columbia Consumer Videos (CCV) [24] and TRECVID MEDTest 14 (MED14) [33].

In addition, we also configured the EVVE as a small recognition dataset with 13 events. For each event, we set the query video as the test data (620 videos), and treat the ground truth in the dataset as the training data. We report the top-1 classification accuracy for performance evaluation.

The Columbia Consumer Videos (CCV) dataset [24] contains 9,317 YouTube videos of 20 classes. We follow the protocol defined in [24], with a training set of 4,659 videos and a test set of 4,658 videos. The TRECVID MEDTest 14 (MED14) [33] is one of the most challenging datasets for event recognition containing 20 complex events. In the training section, there are 100 positive exemplars per event, and all events share negative exemplars with about 5,000 videos. The test section contains approximately 23,000 videos.

For these two datasets, mAP is used as the evaluation metric of event recognition according to the NIST standard

4. More sophisticated recounting inferences can be implemented by computing conditional heat maps based on individual memory layers as we trace the reasoning engine through the layers.

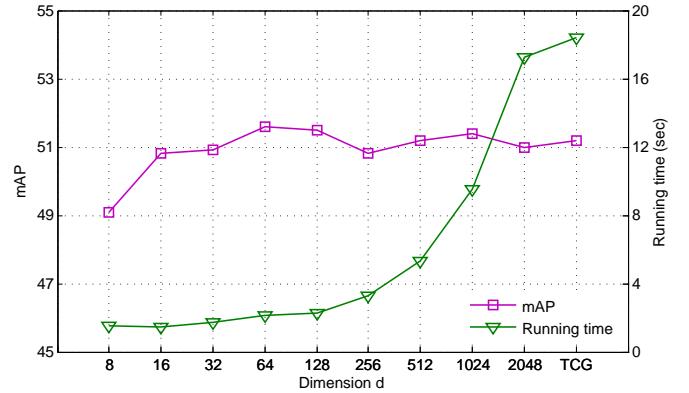


Fig. 5. The average running time of learning epitomes for EVVE dataset with different d and the corresponding retrieval performance (mAP).

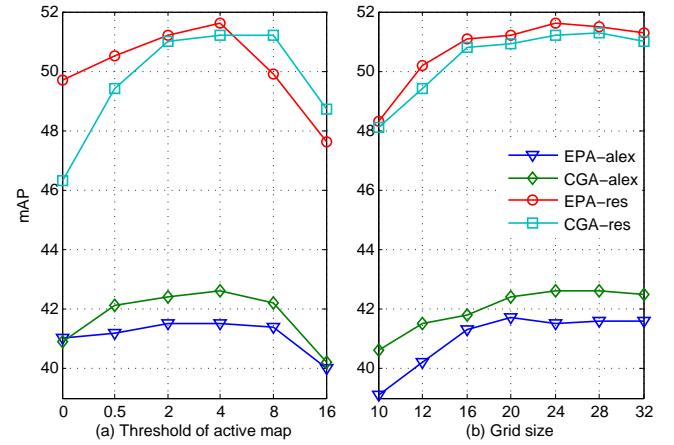


Fig. 6. The influence of different (a) threshold τ of active map and (b) grid sizes E for Counting grid aggregation (CGA) and Epitome aggregation (EPA). alex and res denote two CNN models, *i.e.*, AlexNet and ResNet.

[33]. Since no ground truth is available for the recounting task, we only provide a user study and qualitative analysis for such results.

4.2 Implementation details

Frame-level descriptor. Given an input video, we sample 5 frames per second (5 fps) to extract the CNN features. We explore various pre-trained CNN models, including AlexNet [27], VGG [42] and ResNet-50 [18]. We adopt the output from the last convolutional layer (after ReLU) of these models as the frame descriptors. The CNN feature maps are down-sampled to 4×4 with linear interpolation to fit the TCG (we set $S = 4 \times 4$ in TCG for computational efficiency). For the epitome model, the feature maps are sampled to 8×8 to fit the window size W . In addition, we also average all the frame descriptors over the video (sum-aggregation), as the baseline to evaluate our framework.

Post-processing. For the baseline video representation, we apply the same post-processing strategy as in [1], [16], *i.e.*, the representation vector of a video is first l_2 -normalized, and then whitened using PCA [19] and l_2 -normalized again. For the imprint descriptors on the video

TABLE 2

Comparison with sum-aggregation on EVVE dataset. Sum-, CGA- and EPA- denote sum-aggregation, counting grid aggregation, epitome aggregation, respectively. alex and res denote two CNN models, AlexNet and ResNet-50. For ResNet based representation, the vectors dimension are reduced to 1024 with PCA-whitening. (alex+res) denotes the concatenated vector.

Representation	Dim.	mAP
Sum-alex	256	38.3
Sum-res	1024	46.6
Sum-(alex+res)	1280	47.3
CGA-alex	256	42.6
CGA-res	1024	51.2
CGA-(alex+res)	1280	52.3
EPA-alex	256	41.5
EPA-res	1024	51.6
EPA-(alex+res)	1280	52.1

imprint, power normalization ($\alpha = 0.2$) shows better results than l_2 -normalization in our experiments. Therefore, after feature alignment, the imprint descriptors are first power normalized, then PCA-whitened and l_2 -normalized.

Re-ranking methods for event retrieval. For the event retrieval task on the EVVE dataset, we also employ two variants of query expansion method presented by Douze *et al.* [12]: Average Query Expansion (AQE) and Difference of Neighborhood (DoN). In our experiments, we set $N_1 = 10$ for AQE and $N_1 = 10, N_2 = 2000$ for DoN.

Training details for the reasoning network. The reasoning network (RNet) is trained with stochastic gradient descent (SGD). The initial learning rate is $\beta = 0.025$, which is then annealed every 5 epochs by $\beta/2$ until 20 epochs are finished. All weights are initialized randomly from a Gaussian distribution with zero mean and $\sigma = 0.05$. The weights are shared among different memory layers. The batch size is 128 and the gradients with an l_2 norm larger than 20 are rescaled to norm 20 during the training step.

4.3 Complexity analysis

The most time consuming step is constructing video imprint for input videos. As shown in Figure 5, the average running time of TCG (with ResNet features) for EVVE (about 1200 frames per video) implemented on the GPU platform (K40 with MATLAB parallel computing toolbox) is about 18 seconds. As discussed in [34], [35], with efficient use of cumulative sums, the computational complexity of learning TCG with the EM algorithm grows at most linearly with the product of counting grid size and the tessellation sections $E_x \cdot E_y \cdot S_x \cdot S_y$.

For the epitome, the computational complexity without the efficient two-step scheme for D -dimensional input feature maps and EM iteration times n ($n \approx 30$) is $O(nD)$. With the proposed efficient two-step scheme, the computational complexity becomes $O(nd + D)$. When $d \ll D$, the efficient two-step scheme can significantly accelerate the learning stage. Figure 5 shows the average running time of learning epitomes for the EVVE dataset with different d and the corresponding retrieval performance (mAP). When $d = 64$, without any performance losses, the efficient two-

step scheme achieves almost an order of magnitude speed-up.

4.4 Evaluation results on event retrieval

4.4.1 Parameter analysis

Threshold τ of the active map. Figure 6 (a) shows the retrieval performance with different threshold τ used for the active map construction. We can observe that increasing τ helps filter out some very short shots (with small number of frames) which are usually irrelevant. We set $\tau = 4$ in the subsequent experiments.

Grid size E. To evaluate the influence of the grid size E of the TCG and epitome models, we first fix the window size ($W = 8 \times 8$) of TCG, epitome, and tessellation size ($S = 4 \times 4$) of TCG. Subsequently, we choose 7 different counting grid sizes to perform the feature alignment with the epitome and TCG model, respectively. The performance with regard to size choices is presented in Figure 6 (b). No further improvement can be obtained when $E_x = E_y > 24$. Therefore, the size of the grid is fixed at 24 for all following experiments.

4.4.2 Comparison with sum-aggregation

We refer to our unsupervised flow (combining the feature alignment and aggregation steps) on ER3 as counting grid aggregation (CGA) for TCG based video imprint and epitome aggregation (EPA) for epitome based video imprint. Table 2 shows the retrieval performance compared with baselines. We evaluate the CGA and EPA on two different CNN models, AlexNet [27] and ResNet-50 [18]. Figure 7 shows the retrieval performance for each event class of EVVE dataset. The IDs of events are the same with [38]. We can see that the video representation based on ResNet-50 demonstrates superior performance in most event categories compared with AlexNet. And the proposed aggregation methods can further boost the retrieval performance.

As shown in Table 2, compared with Sum-aggregation, CGA and EPA both obtain better retrieval performance with the benefits from feature alignment step that can suppress the redundancy among frames. In addition, consistent improvement can be observed for different CNN models (AlexNet and ResNet-50). After concatenating the video representation vectors from these two CNN models, the retrieval performance can be further improved, mAP = 52.3 for CGA and mAP = 52.1 for EPA.

4.4.3 Comparison with state-of-the-arts

In Table 3, we can see that the sum-aggregation with CNN features already outperforms previous work [12], [37], [38]. After merging with 100K distractors, the mAPs of CGA-(alex+res) and EPA-(alex+res) achieve 42.9 and 43.9 respectively, which are still better than the baseline (mAP = 38.7) and previous work [12], [37], [38]. For fair comparison, we also reimplement previous aggregation methods (CTE, TMK and SHP) based on the same CNN features with CGA and EPA. Note that MMV is in fact the sum-aggregation based on the multi-VLAD frame feature [38], which employs the same aggregation method with our baseline method.

As shown in Table 3, frame-level descriptors have a huge impact on performance. CNN feature-based aggregation methods achieve better retrieval performance than

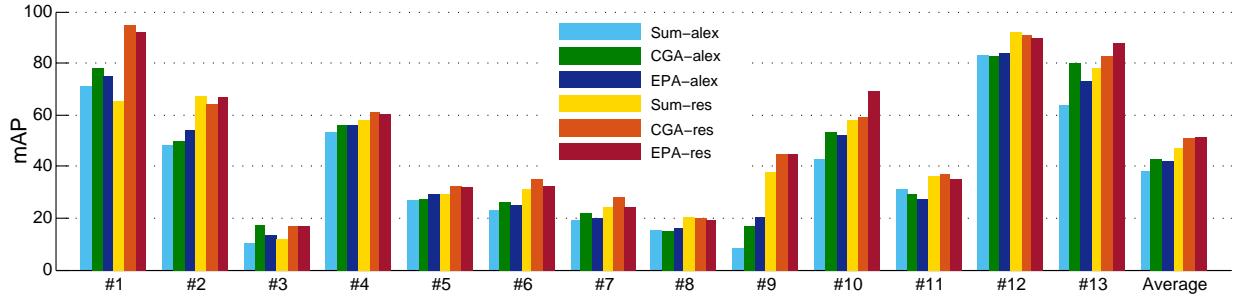


Fig. 7. Retrieval performance (mAP) per event.

TABLE 3
Retrieval performance compared with other methods. AQE and DoN denote the two Re-ranking methods.

Method	Dim.	EVVE			EVVE+100K		
		no QE	AQE	DoN	no QE	AQE	DoN
MMV [38]	512	33.4	—	—	22.0	—	—
CTE [38]	—	35.2	—	—	20.2	—	—
MMV+CTE [38]	—	37.6	—	—	25.4	—	—
TMK [37]	66560	33.5	36.1	41.3	25.4	—	34.7
SHP [12]	16384	36.3	38.9	44.0	26.5	30.1	33.1
CTE-(alex+res)	—	44.7	—	—	40.1	—	—
TMK-(alex+res)	42240	46.7	51.8	53.0	40.9	48.4	48.0
SHP-(alex+res)	40960	48.7	54.0	55.4	41.5	48.7	48.9
Sum-(alex+res)	1280	47.3	53.1	55.2	38.7	45.8	47.1
CGA-(alex+res)	1280	52.3	58.5	60.1	42.9	50.4	52.7
EPA-(alex+res)	1280	52.1	59.3	59.9	43.9	53.2	53.8

their original handcrafted feature-based implementations. However, based on the same frame-level descriptors, our aggregation methods still obtain better results compared with existing methods. In addition, previous aggregation methods usually lead to much higher dimension in the video representation (which may reduce the efficiency of the retrieval task). Based on the initial retrieval results, the query expansion can further boost the performance. We achieve 8.5% and 8.9% improvement compared with previous result ($mAP = 55.4$ for SHP [12] combined with CNN feature) and the baseline ($mAP = 55.2$) on the EVVE dataset, respectively. Consistent improvement is also observed with query expansion on the large dataset (EVVE+100K).

4.5 Evaluation results on event recognition

4.5.1 Parameter analysis

Structure of the reasoning network. For the EVVE dataset, we set a softmax layer as the decision network. The video imprint is generated based on the ResNet-50 [18] model and its imprint descriptors are first reduced to 256 dimension with PCA-whitening before being fed to the reasoning network (RNet). For the CCV and MED14 datasets, we add a fully connected layer in front of the softmax layer as the decision network for better performance. Besides the ResNet-50 model, we also evaluate the framework with VGG (16 layers) [42] model on these two datasets. The dimension of the imprint descriptors is set to 1024 and 512 for ResNet-50 and VGG, respectively. For all the datasets,

TABLE 4
Comparison with sum-aggregation and CGA/EPA. Sum-, CGA- and EPA- denote sum-aggregation, counting grid aggregation and epitome aggregation, respectively. RNet- denote the reasoning network. vgg and res denote two CNN model, VGG and ResNet-50. (vgg+res) denotes the later fusion result.

	Method	vgg	res	(vgg+res)
CCV	Sum-	74.3	75.3	78.1
	CGA-	75.7	76.6	79.1
	EPA-	75.2	76.7	78.3
	CG-RNet-	76.7	78.5	79.9
	EP-RNet-	75.8	78.2	79.0
MED14	Sum-	26.0	30.4	32.8
	CGA-	30.5	32.2	33.7
	EPA-	31.2	32.4	34.4
	CG-RNet-	32.8	34.2	36.9
	EP-RNet-	32.6	34.3	36.3

the internal vectors b_i and m_i have the same dimensions with the input imprint descriptors.

Number of memory layers. Figure 8 illustrates the influence of RNet with increased hops on the EVVE dataset. To make a fair comparison, we employ the same decision network as classifier for the baselines and the output from RNet. We compare with two representations, one is the video representation with sum-aggregation, and the other is either the counting grid aggregation (CGA) or the epitome aggregation (EPA) which depends on the feature alignment step. In fact, if we fix the value of the weights

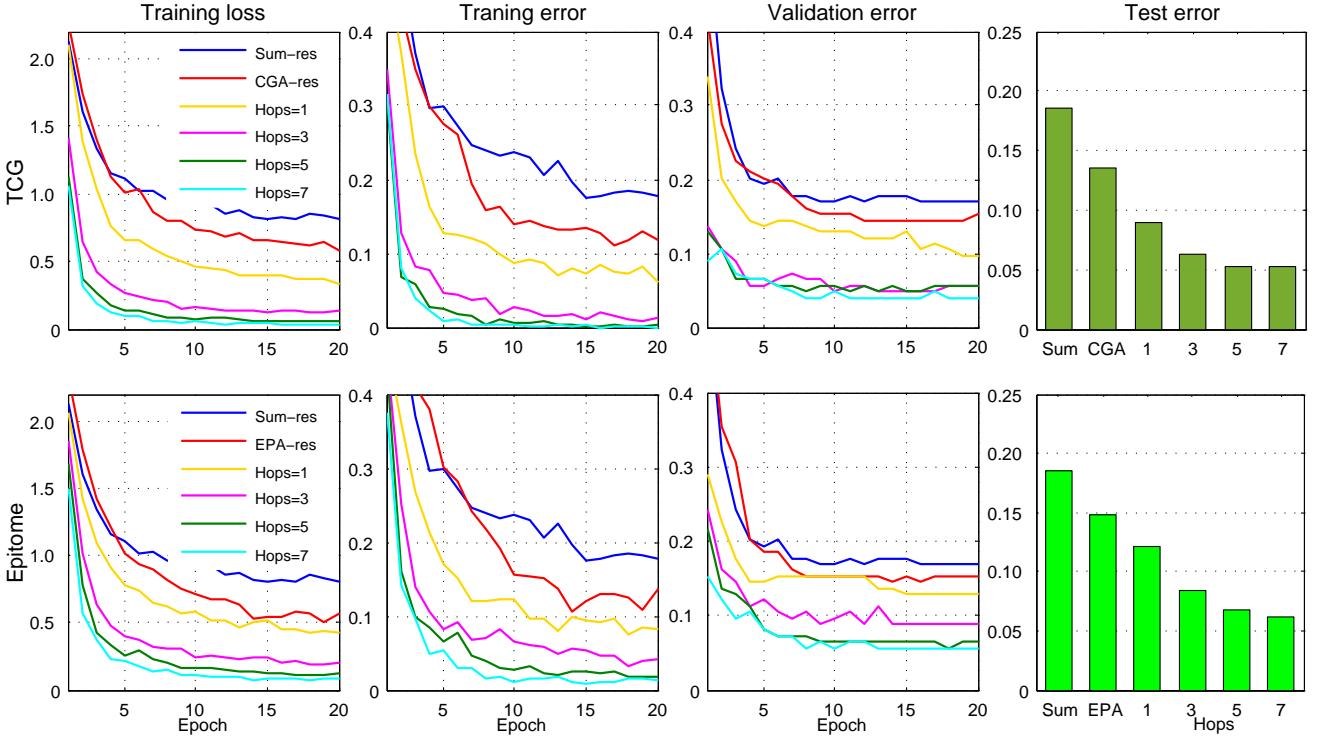


Fig. 8. The influence of RNet with increased hops on EVVE dataset. Best viewed in color.

TABLE 5

Comparison with other methods. MA+CG/EP-RNet-(vgg+res) denote the results fused with audio and motion information using adaptive fusion method [55]. IDT+CG/EP-RNet-(vgg+res) denote the results fused with improved dense trajectories [52].

	Method	mAP	Recounting
CCV	Lai <i>et al.</i> [28]	43.6	✓
	Jiang <i>et al.</i> [24]	59.5	✗
	Wu <i>et al.</i> [54]	70.6	✗
	Nagel <i>et al.</i> [32]	71.7	✗
	Wu <i>et al.</i> [55]	84.9	✗
MED14	CG-RNet-(vgg+res)	79.9	✓
	EP-RNet-(vgg+res)	79.0	✓
	MA+CG-RNet-(vgg+res)	87.1	✓
	MA+EP-RNet-(vgg+res)	86.8	✓
	IDT [33], [52]	27.6	✗
MED14	Gan <i>et al.</i> [13]	33.3	✓
	Xu <i>et al.</i> [57]	36.8	✗
	Zha <i>et al.</i> [60]	38.7	✗
	CG-RNet-(vgg+res)	36.9	✓
	EP-RNet-(vgg+res)	36.3	✓
	IDT+CG-RNet-(vgg+res)	40.2	✓
	IDT+EP-RNet-(vgg+res)	39.8	✓

map equal to the active map, the RNet will reduce to the CGA or EPA, *i.e.*, the unsupervised flow in Figure 1. We can see that CGA/EPA provides better performance than sum-aggregation, and the RNet can further refine the video representation and lead to better recognition accuracy than the two baselines. In addition, the gain is also increased with more hops. Consistent gains are observed on both CCV and MED14 datasets. We set the hops = 3 in the following

experiments on the CCV and MED14 datasets.

4.5.2 Performance on CCV and MED14

Table 4 shows the recognition performance (mAP) of RNet and baseline methods. With the benefit from re-weighting the video imprint, the RNet achieves better results on both CCV (mAP = 79.9/79.0) and MED14 (mAP = 36.9/36.3) datasets compared with sum-aggregation and CGA/EPA. In addition, on the CCV dataset, we also employ the same strategy as [55] to combine motion and audio features with our appearance-based representation. As shown in Table 5, the fusion strategy further boosts the recognition performance (mAP = 87.1/86.8). On MED14 dataset, the proposed method sets a new performance record (mAP = 40.2) after fusing motion features (IDT) in a similar way to [60], with the additional convenience of simultaneously providing recounting results.

4.6 Evaluation results on event recounting

Influence of average pooling. In contrast to the original memory networks [43], we add an average pooling layer inside the memory layer, which takes advantage of the spatial organization of the information in the video imprint. Figure 9 demonstrates the influence of adding the average pooling layer. We can see that the recounting maps are smoother and more reasonable, especially for TCG based video imprint. In addition, with benefits from finer resolution of input feature maps, the epitome based video imprint can hold more spatial information, and the recounting results are more reasonable than TCG based video imprint.

User study. The evaluation of event recounting is not easy since there is no ground truth information. To assess

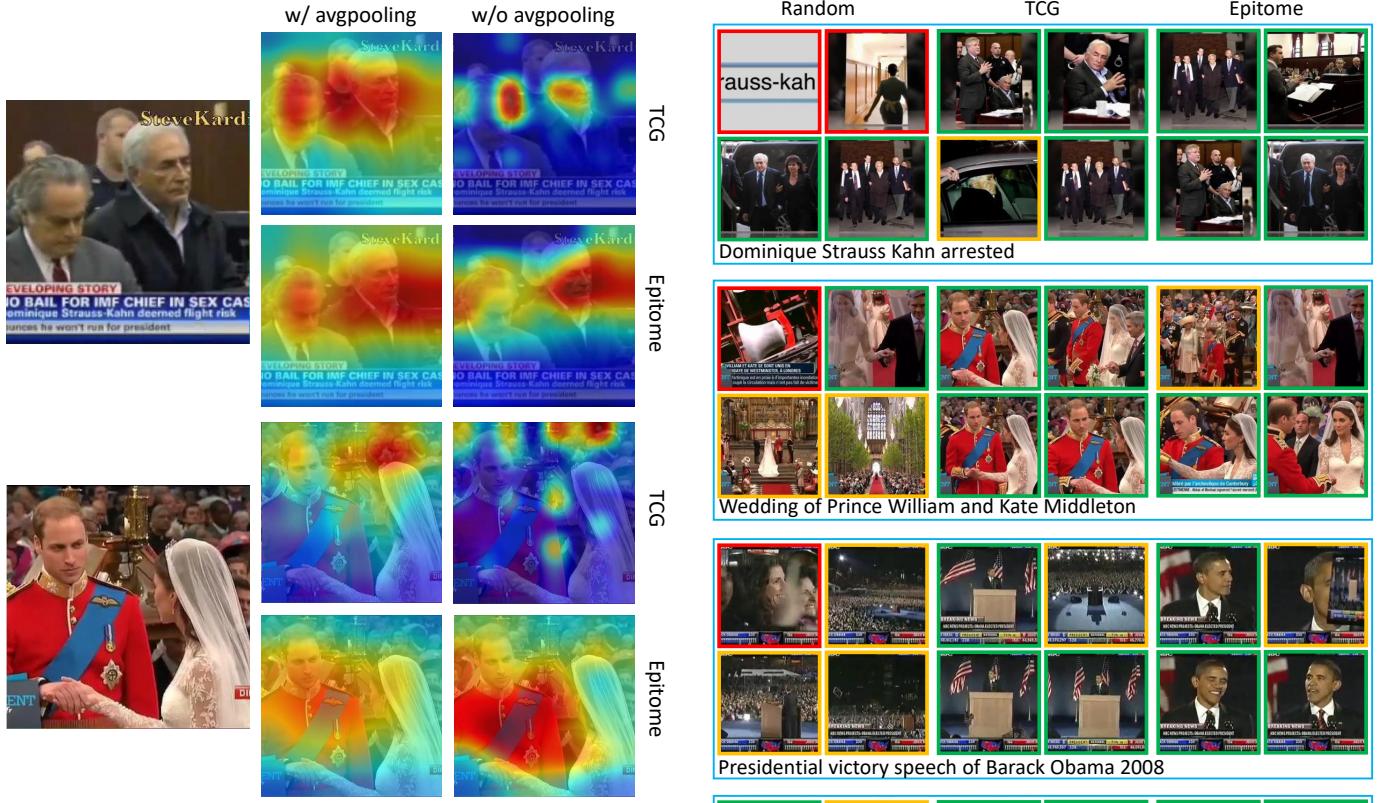


Fig. 9. Influence of the average pooling layer in RNet. The middle column shows the recounting map of RNet. The right column shows the recounting map with avg-pooling layer removed. Best viewed in color.

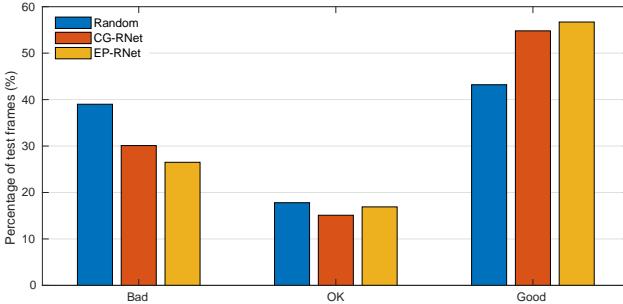


Fig. 10. The statistical results of user study. The average percentage of key frames belong to **Bad**, **OK** and **Good** are shown with different color bar. Best viewed in color.

the quality of our recounting results, we randomly sample 200 videos from 620 test videos of the EVVE dataset for user study. First, for each test video, we sample 4 key frames either randomly or based on the importance score from the recounting map \mathbf{R} (3 groups in total: randomly selected, based on important scores from TCG or Epitome). Then, we invited 50 users to score the key frames with **{Bad, OK, Good}** based on the relevance with the labeled event. Here, **Bad** means that the selected frame is irrelevant with its event label, *i.e.*, users cannot recognize the event category by the selected frame. **OK** means that users can identify the event category by the selected frame with reasonable amount of guesswork. **Good** means that the event category



Fig. 11. Examples of the key frames for user study. Each blue box contains the key frames proposed from a test video with three strategies, *i.e.*, random selection, recounting results based on TCG or epitome. Each strategy proposes four key frames to represent the video. The scores ranked by the users are shown with different color boxes, red for **Bad**, yellow for **OK** and green for **Good**. Best viewed in color.

can be identified beyond doubt by the selected frame and users are highly confident with the judgment.

The statistical results are shown in Figure 10. It presents the average percentage of key frames belong to **Bad**, **OK** or **Good**, respectively. Compared with random selection, the key frames proposed with recounting map are more likely to be relevant with the corresponding event, the percentage of bad proposals is reduced from 39% to 26%. Note that epitome based recounting results are slightly better than TCG due to richer spatial information. Figure 11 shows the examples of key frames for user study. Each blue box contains a test video represented by 4 key frames. The key frames are proposed with three strategies, *i.e.*, random selection, recounting based on TCG or epitome. The scores ranked by the users are shown with different color boxes.

Recounting map. Figure 12 illustrates some examples of the recounting results. The heat map is used to visualize the recounting map (the map is rescaled to the same size with

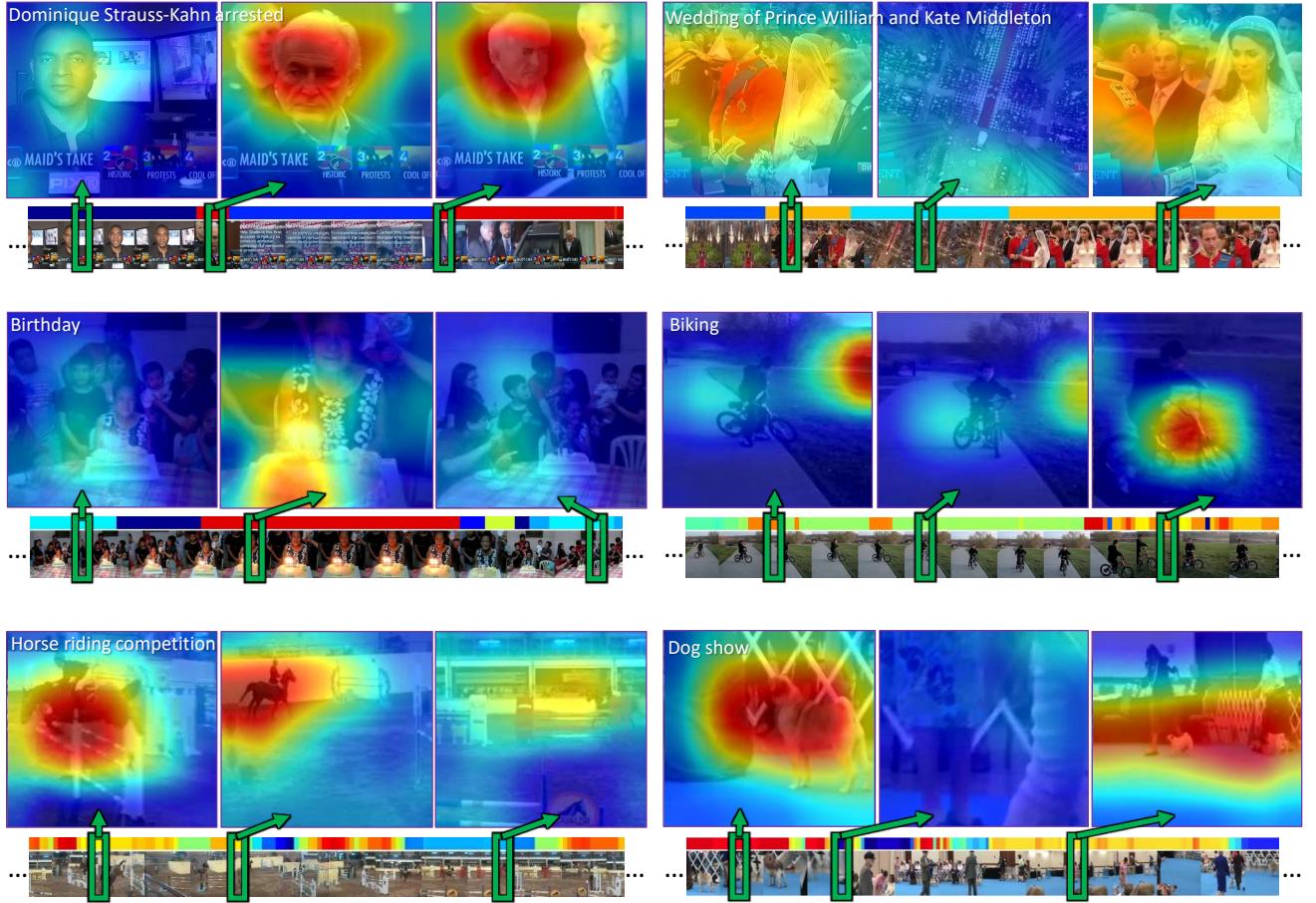


Fig. 12. Examples of event recounting results. We use heat map to indicate the recounting map. The key areas related to the event in each frame is painted with red color. The importance score which is computed by the sum of recounting map is shown with color bar (red for important frames) upon the video frame flow. Best viewed in color.

the frame). Since no ground truth recounting is available, we can only provide some examples as shown in Figure 12. We can see that our recounting process can not only provide the importance score of each frame, but also indicate the most relevant areas inside each frame. However, due to the coarse resolution of the input feature maps, the spatial-level recounting results are also very coarse. Nevertheless, the recounting heat map may be used as a good prior for other post-processing methods, *e.g.*, object segmentation.

5 CONCLUSION

In this paper, we propose an efficient and effective event retrieval, recognition and recounting framework (ER3), based on our proposed video imprint representation. In the proposed framework, a dedicated feature alignment module is incorporated for redundancy removal across frames to produce the compact intermediate tensor representation, *i.e.*, the video imprint. Subsequently, the video imprint is processed by a reasoning network for event recognition/recounting, and by a feature aggregation module for event retrieval. Thanks to the attention mechanism inspired by the memory networks in language modeling, the proposed reasoning network is capable of simultaneous event recognition and event recounting. With the event retrieval

task, the compact video representation aggregated from the video imprint contributes to better retrieval results.

ACKNOWLEDGMENTS

This work was supported partly by National Key R&D Program of China Grant 2017YFA0700800, National Natural Science Foundation of China Grants 61629301, 61773312, 91748208, and 61503296, China Postdoctoral Science Foundation Grants 2017T100752 and 2015M572563.

REFERENCES

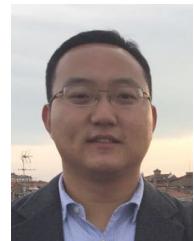
- [1] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1269–1277, 2015.
- [2] M. Baillie and J. M. Jose. Audio-based event detection for sports video. In *Proc. Int'l Conf. Image and Video Retrieval*, pages 300–309, 2003.
- [3] L. Baraldi, M. Douze, R. Cucchiara, and H. Jégou. LAMV: Learning to align and match videos with kernelized temporal layers. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018.
- [4] S. Bhattacharya, M. M. Kalayeh, R. Sukthankar, and M. Shah. Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2235–2242, 2014.
- [5] H. Bilen, B. Fernando, É. Gavves, and A. Vedaldi. Action recognition with dynamic image networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2017.

- [6] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 961–970, 2015.
- [7] L. Cao, Y. Mu, A. Natsev, S.-F. Chang, G. Hua, and J. R. Smith. Scene aligned pooling for complex video recognition. In *Proc. European Conf. Computer Vision*, pages 688–701, 2012.
- [8] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intelligent Systems and Technology*, 2(3):27, 2011.
- [9] X. Chang, Y.-L. Yu, Y. Yang, and A. G. Hauptmann. Searching persuasively: Joint event detection and evidence recounting with limited supervision. In *Proc. ACM Int'l Conf. Multimedia*, pages 581–590, 2015.
- [10] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing. Semantic pooling for complex event analysis in untrimmed videos. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 39(8):1617–1632, 2017.
- [11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 886–893, 2005.
- [12] M. Douze, J. Revaud, C. Schmid, and H. Jégou. Stable hyperpooling and query expansion for event detection. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1825–1832, 2013.
- [13] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann. DevNet: A deep event network for multimedia event detection and evidence recounting. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2568–2577, 2015.
- [14] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei. You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 923–932, 2016.
- [15] Z. Gao, G. Hua, D. Zhang, N. Jojic, L. Wang, J. Xue, and N. Zheng. ER3: A unified framework for event retrieval, recognition and recounting. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2253–2262, 2017.
- [16] Z. Gao, J. Xue, W. Zhou, S. Pang, and Q. Tian. Democratic diffusion aggregation for image retrieval. *IEEE Trans. Multimedia*, 18(8):1661–1674, 2016.
- [17] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Proc. European Conf. Computer Vision*, pages 392–407, 2014.
- [18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [19] H. Jégou and O. Chum. Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening. In *Proc. European Conf. Computer Vision*, pages 774–787, 2012.
- [20] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3304–3311, 2010.
- [21] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.
- [22] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah. High-level event recognition in unconstrained videos. *Int'l journal of multimedia information retrieval*, 2(2):73–101, 2013.
- [23] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2017.
- [24] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *Proc. Int'l Conf. Multimedia Retrieval*, pages 29–37, 2011.
- [25] N. Jojic, B. J. Frey, and A. Kannan. Epitomic analysis of appearance and shape. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 34–41, 2003.
- [26] N. Jojic and A. Perina. Multidimensional counting grids: Inferring word order from disordered bags of words. In *Proc. Conf. Uncertainty in Artificial Intelligence*, pages 547–556, 2011.
- [27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Conf. Neural Information Processing Systems*, pages 1097–1105, 2012.
- [28] K.-T. Lai, X. Y. Felix, M.-S. Chen, and S.-F. Chang. Video event detection by inferring temporal instance labels. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2251–2258, 2014.
- [29] K.-T. Lai, D. Liu, M.-S. Chen, and S.-F. Chang. Recognizing complex events in videos by learning key static-dynamic evidences. In *Proc. European Conf. Computer Vision*, pages 675–688, 2014.
- [30] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l Journal of computer vision*, 60(2):91–110, 2004.
- [31] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. G. Hauptmann. Complex event detection via multi-source video attributes. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2627–2633, 2013.
- [32] M. Nagel, T. Mensink, C. G. Snoek, et al. Event fisher vectors: Robust encoding visual diversity of visual streams. In *Proc. British Machine Vision Conf.*, pages 6–18, 2015.
- [33] P. Over, J. Fiscus, G. Sanders, D. Joy, M. Michel, G. Awad, A. Smeaton, W. Kraaij, and G. Quénot. TRECVID 2014—an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proc. TRECVID*, pages 52–74, 2014.
- [34] A. Perina and N. Jojic. Image analysis by counting on a grid. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1985–1992, 2011.
- [35] A. Perina and N. Jojic. Capturing spatial interdependence in image features: the counting grid, an epitomic representation for bags of features. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 37(12):2374–2387, 2015.
- [36] F. Perronnin and D. Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3743–3752, 2015.
- [37] S. Poullot, S. Tsukatani, A. Phuong Nguyen, H. Jégou, and S. Satoh. Temporal matching kernel with explicit feature maps. In *Proc. ACM Int'l Conf. Multimedia*, pages 381–390, 2015.
- [38] J. Revaud, M. Douze, C. Schmid, and H. Jégou. Event retrieval in large video collections with circulant temporal encoding. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2459–2466, 2013.
- [39] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *Int'l Journal of computer vision*, 105(3):222–245, 2013.
- [40] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.
- [41] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. Conf. Neural Information Processing Systems*, pages 568–576, 2014.
- [42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [43] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *Proc. Conf. Neural Information Processing Systems*, pages 2440–2448, 2015.
- [44] C. Sun and R. Nevatia. ACTIVE: Activity concept transitions in video event classification. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 913–920, 2013.
- [45] C. Sun and R. Nevatia. DISCOVER: Discovering important segments for classification of video events and recounting. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2569–2576, 2014.
- [46] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.
- [47] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Proc. Conf. Computer Graphics and Interactive Techniques*, pages 251–258, 1997.
- [48] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classes. In *Proc. European Conf. Computer Vision*, pages 776–789, 2010.
- [49] D. Tran, J. Yuan, and D. Forsyth. Video event detection: From subvolume localization to spatiotemporal path search. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 36(2):404–416, 2014.
- [50] C.-Y. Tsai, M. L. Alexander, N. Okwara, and J. R. Kender. Highly efficient multimedia event recounting from user semantic preferences. In *Proc. Int'l Conf. Multimedia Retrieval*, pages 419–427, 2014.
- [51] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3169–3176, 2011.

- [52] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 3551–3558, 2013.
- [53] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.
- [54] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue. Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In *Proc. ACM Int'l Conf. Multimedia*, pages 167–176, 2014.
- [55] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue. Multi-stream multi-class fusion of deep networks for video classification. In *Proc. ACM Int'l Conf. Multimedia*, pages 791–800, 2016.
- [56] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proc. ACM Int'l Conf. Multimedia*, pages 461–470, 2015.
- [57] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative CNN video representation for event detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1798–1807, 2015.
- [58] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation networks for video face recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2492–2495, 2017.
- [59] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.
- [60] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained CNN architectures for unconstrained video classification. In *Proc. British Machine Vision Conf.*, pages 60–73, 2015.
- [61] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 2117–2126, 2017.
- [62] Q. Zhang and G. Hua. Multi-view visual recognition of imperfect testing data. In *Proc. ACM Int'l Conf. Multimedia*, pages 561–570, 2015.



Zhanning Gao received the B.S. degree in automatic control engineering from Xi'an Jiaotong University, Xi'an, China, in 2012. He is currently a Ph.D. candidate in Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. He was a research intern in Visual Computing Group in Microsoft Research Asia from 2015 to 2017. His research interests include compact image/video representation, large scale content based multimedia retrieval and complex event video analysis.



for web images and videos. He is the author of more than 20 peer reviewed publications in prestigious international journals and conferences. He is a member of the IEEE.



Nebojsa Jojic received the PhD degree from the University of Illinois at Urbana-Champaign in 2001, where he received a Microsoft Fellowship in 1999 and a Robert T. Chien Excellence in Research award in 2000. He has been a researcher at Microsoft Research in Redmond, Washington, since 2000. He has published more than 100 papers in the areas of computer vision, machine learning, signal processing, computer graphics, and computational biology.



Zhenxing Niu received the Ph.D. degree in Control Science and Engineering from Xidian University, Xi'an, China, in 2012. From 2013 to 2014, he was a visiting scholar with University of Texas at San Antonio, Texas, USA. He is a Researcher at Alibaba Group, Hangzhou, China. Before joining Alibaba Group, he is an Associate Professor of School of Electronic Engineering at Xidian University, Xi'an, China. His research interests include computer vision, machine learning, and their application in object discovery and localization. He served as PC member of CVPR, IJCV, and ACM Multimedia. He is a member of the IEEE.



Nanning Zheng (SM'94-F'06) graduated in 1975 from the Department of Electrical Engineering, Xi'an Jiaotong University (XJTU), received the ME degree in Information and Control Engineering from Xi'an Jiaotong University in 1981, and a Ph. D. degree in Electrical Engineering from Keio University in 1985. He is currently a Professor and the director with the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, computational intelligence, and hardware implementation of intelligent systems. Since 2000, he has been the Chinese representative on the Governing Board of the International Association for Pattern Recognition. He became a member of the Chinese Academy Engineering in 1999. He is a fellow of the IEEE.



Gang Hua (M'03-SM'11) was enrolled in the Special Class for the Gifted Young of Xi'an Jiaotong University (XJTU) in 1994 and received the B.S. degree in Automatic Control Engineering from XJTU in 1999. He received the M.S. degree in Control Science and Engineering in 2002 from XJTU, and the Ph.D. degree in Electrical Engineering and Computer Science at Northwestern University in 2006. He is currently a Principle Researcher/Research Manager at Microsoft Research. Before that, he was an Associate Professor of Computer Science at Stevens Institute of Technology. He also held an Academic Advisor position at IBM T. J. Watson Research Center between 2011 and 2014. He was a Research Staff Member at IBM Research T. J. Watson Center from 2010 to 2011, a Senior Researcher at Nokia Research Center, Hollywood from 2009 to 2010, and a Scientist at Microsoft Live Labs Research from 2006 to 2009. He is currently an Associate Editor in Chief for CVIU, and Associate Editors for IJCV, IEEE T-IP, IEEE T-CSVT, IEEE Multimedia, and MVA. He also served as the Lead Guest Editor on two special issues in TPAMI and IJCV, respectively. He is a program chair of CVPR'2019&2022. He is an area chair of CVPR'2015&2017, ICCV'2011&2017, ICIP'2012&2013&2016, ICASSP'2012&2013, and ACM MM 2011&2012&2015&2017. He is the author of more than 150 peer reviewed publications in prestigious international journals and conferences. He holds 19 issued US patents and has 20 more US patents pending. He is the recipient of the 2015 IAPR Young Biometrics Investigator Award for his contribution on Unconstrained Face Recognition from Images and Videos, and a recipient of the 2013 Google Research Faculty Award. He is an IAPR Fellow, an ACM Distinguished Scientist, and a senior member of the IEEE.