ML-HDP: A Hierarchical Bayesian Nonparametric Model for Recognizing Human Actions in Video

Nguyen Anh Tu, *Student Member, IEEE,* Thien Huynh-The, *Student Member, IEEE,* Kifayat Ullah Khan, *Member, IEEE,* and Young-Koo Lee, *Member, IEEE,*

***Abstract*— Action recognition from videos is an important area of computer vision research due to its various applications, ranging from visual surveillance to human-computer interaction. To address action recognition problems, this paper presents a framework that jointly models multiple complex actions and motion units at different hierarchical levels. We achieve this by proposing a generative topic model, namely Multi-label Hierarchical Dirichlet Process (ML-HDP). The ML-HDP model formulates the co-occurrence relationship of actions and motion units, and enables highly accurate recognition. In particular, our topic model possesses the three-level representation in action understanding, where low-level local features are connected to high-level actions via mid-level atomic actions. This allows the recognition model to work discriminatively. In our ML-HDP, atomic actions are treated as latent topics and automatically discovered from data. In addition, we incorporate the notion of class labels into our model in a semi-supervised fashion to effectively learn and infer multi-labeled videos. Using discovered topics and inferred labels, which are jointly assigned to local features, we present the straightforward methods to perform three recognition tasks including: action classification, joint classification and segmentation of continuous actions, and spatiotemporal action localization. In experiments, we explore the use of three different features and demonstrate the effectiveness of our proposed approach for these tasks on four public datasets: KTH, MSR-II, Hollywood2, and UCF101.**

***Index Terms*—Action recognition, action segmentation, action localization, topic modeling, Dirichlet process.**

I. Introduction

NOWADAYS, the rapid growth of camera devices and social media sharing platforms (e.g., Youtube and Facebook) has created a vast volumes of video data. This makes human action recognition become an active field in the computer vision research. The ability of understanding complex actions plays an important role in many high-impact societal applications including smart surveillance, web-video search and retrieval, patient monitoring, sports analysis, and human-computer interaction. Many researches have been carried out in the literature, but action recognition still remains challenging due to several reasons: videos taken with background clutter and motion; large variety of lighting conditions and viewpoints; limited quantities of labeled data for learning process; and large intra-class variability versus small inter-class variability. Over the past decade, a lot of works have leveraged the local space-time features in handling these challenges. Different feature types have been proposed such as Space-Time Interest Points (STIP) [1] and Dense Trajectories (DT) [3], [4], which are extracted directly from a video without tracking human bodies and can be then input to generative or discriminative recognition models.

Using the local features, many conventional approaches [2], [5] have focused on solving the problem of single action recognition, where the boundaries of individual actions are known in advance. In other words, each video in this case contains only one action performed by a person. Consequently, action recognition problem turns into a classification problem that assigns an action label to the pre-segmented video. Typically, a video is represented as a histogram over the visual vocabulary. Then, histograms of labeled training data are fed into a classifier to learn a model for each action class. However, if we only treat action recognition as classification problem, it seems unrealistic for many applications such as surveillance with continuous data streams, where the boundaries of individual actions in video are usually unknown.

Moving beyond simple action classification, many attempts have been made to deal with more realistic and challenging multi-action recognition, which aim to recognize multiple actions present in a video. In the context of this paper, we define two types of multi-action recognition problems: segmentation and localization. First, continuous action segmentation is to segment and classify a sequence of unknown actions performed through time within a video. Most of previous studies [6], [7] have usually treated this problem as two separate tasks, i.e. segmentation and classification, but this strategy may cause the significant loss of information related to actions [8]. Besides, these approaches require fully annotated training videos with known segmentation and action labels, which is time-consuming and laborious in real-world situations. Second, action localization is to classify actions and identify their space-time segments on a video. Different from continuous action segmentation, there may exist overlaps between these segments relevant to the cases that different people perform multiple actions at the same time. Traditionally, to handle this problem, sliding window-based approaches [9], [14], which perform the exhaustive search of potential action locations, have been widely applied. However, it is generally recognized that such search strategy is costly due to huge search space of video patterns. Moreover, sliding window-based approaches are discriminative in nature, since they localize each action

Nguyen Anh Tu, Thien Huynh-The, Kifayat Ullah Khan, and Young-Koo Lee work in the Department of Computer Science and Engineering, Kyung Hee University (Global Campus), Korea. E-mail: tunguyen@khu.ac.kr; thienht@oslab.khu.ac.kr; kualizai@khu.ac.kr; yklee@khu.ac.kr.

class independently from the others, making them unaware of co-existence of different actions within a video. This leads to increasing uncertainty for each action class.

In this paper, we develop a novel framework based on the bag-of-words representation to overcome all aforementioned limitations, and enable more accurate recognition. Specifically, we aim to jointly model high-level actions and lower-level motion units in a generative model to explicitly exploit the relationship of their co-occurrence at different hierarchical levels. To this end, motivated by the great success of topic modeling in natural language processing [10], [11], we propose a probabilistic topic model, namely Multi-label Hierarchical Dirichlet Process (ML-HDP), to learn human actions. Our topic model is constructed based on the three-level representation as follows: (1) High-level actions (e.g., running) are modeled using combinations of co-occurring atomic actions (e.g., leg motions, hand motions, near-field running, and far-field running). (2) Atomic actions, which are treated as latent topics, are modeled using combinations of co-occurring local features those represent multiple visual events such as position, velocities, and appearance. (3) At the lowest level, local features are quantized via the learned codebook to be formed as visual words. Compared with previous approaches, our topic model shares two attractive benefits. First, by jointly modeling multiple actions, each action known to occur in a video can help to distinguish the appearance of the others. Second, due to the use of a generative model, our framework is naturally suited for weakly-supervised settings. The fully labeled data, such as known segmentation with action order for continuous action segmentation, are not strongly required in our framework. During the learning process, using only weak labels to indicate the presence of actions makes our approach more flexible to practical applications with little human labor.

According to the use of topic model for action recognition, our approach is inspired by the work of [36], where they learn human actions with LDA [10] and pLSA [30] models. Their approach treats high-level actions as topics and performs several recognition tasks based on topic assignments of visual words. However, these parametric models are unsupervised and only produce two-level action representation, which significantly limits the recognition performance under challenging scenarios such as small inter-class variation. Our ML-HDP is also related to the HDP model proposed in [34], which builds the nonparametric model in terms of three-level structure to group similar clips of crowded scenes into clusters. However, their model is also unsupervised and specially designed for clustering task. In particular, they model data with only one cluster label per video clip, and hence it fails to apply their model to solve multi-label problems as aforementioned. Consequently, our first contribution is that atomic actions (i.e., topics) in ML-HDP are shared across the collection of videos and act as mid-level representation to describe simple and co-occurring motion patterns. This helps to fill the semantic gap between low-level features and high-level concepts, and so enables the robust learning. Moreover, due to the non-parametric property of Dirichlet Process [11], a number of these atomic actions are automatically discovered from video data. The second contribution is that we explicitly integrate the notions of action labels into ML-HDP to directly perform learning and inference from multi-labeled videos in a semi-supervised fashion. As a result, each visual word is assigned to an inferred label corresponding to an action instance. Using label assignments allows us to effectively classify single actions as well as jointly classify and segment/localize multiple actions with straightforward methods. More specifically, for multi-action recognition, we employ per-frame representation, which considers temporal ordering of features to assign labels at different time stamps. Finally, three different low-level features including two hand-crafted features and one deep-learned feature are examined in a wide variety of recognition experiments. This further demonstrates the effectiveness of ML-HDP and its compatibility with various types of features.

The remainder of this paper is organized as follows. Section II provides an overview of related works. Section III introduces our framework of action recognition, and describes the details of our proposed methods. Experimental results on benchmark datasets are conducted and discussed in Section IV. Finally, conclusions are presented in Section V.

II. Related Work

In this section, we briefly review research studies that are closely related to our proposed method consisting of: action classification; multiple action recognition, i.e. action segmentation and action localization; and topic model for action recognition.

A. Action classification

Action classification has been extensively studied in the literature, where a lot of works have concentrated on the design of robust video representations. Among those works, the bag-of-words model [15] and its variants (e.g., VLAD [18] and Fisher Vectors [19]) together with local features such as STIPs [1], Cuboid [17], and DTs [4] are the most popular approaches for human action recognition [2], [3], [20]. Despite the success of local feature-based methods, the performance of action classification is usually limited due to the semantic gap between low-level features and high-level concepts. To bridge this semantic gap, more sophisticated models have been proposed by exploiting the hierarchical structure of actions [21], [47], [60] or incorporating mid-level action representation [22], [43], [46]. In particular, Wang et al. [21] employed Latent SVM framework by hierarchically decomposing complex actions into sub-actions. Peng et al. [60] proposed a multi-layer stacked Fisher Vectors encoding method to improve the representation of conventional Fisher Vectors and achieved the excellent performance. Video dynamics have been modeled in [47] at three levels, where they trained models using linear dynamic system and VLAD encoder. Besides modeling hierarchical structures, other approaches have been proposed to mine mid-level action representations, those are then learned with discriminative models. Raptis et al. [22] clustered point trajectories into tentative action parts by the similarity in motion and appearance. Differently, action-exemplars and template-matching were used in [43] and [46], respectively, to extract mid-level features

in a supervised manner. More recently, due to the success of convolutional neural networks (CNN), many researches have focused on developing CNN-based models [48]–[50] to recognize actions. For example, Wang et al. [50] designed the CNN features by combining dense trajectories with two-stream convolutional networks, which are trained with RGB frames and optical flows. Different from existing methods, which are all based on discriminative models, we present a generative model to discover atomic actions, i.e. mid-level representations. However, the difference between our atomic actions and other hierarchical or mid-level representations is that our atomic actions are discovered by capturing the co-occurrence relationship among visual patterns.

B. Multiple action recognition

Compared to action classification problem, which deals with pre-segmented videos, multiple action recognition is more challenging and natural problem in real-world applications. The reason is that we have to handle unsegmented videos with multiple actions. As mentioned earlier, we divide related works on multiple action recognition into two categories: action segmentation and action localization.

Action segmentation: In the first category, we only focus on studies that address the problem of joint classification and segmentation. In this line of work, most approaches [8], [23]–[25] cast action segmentation as a sequence problem by capturing the dependency and transition between classes to facilitate the action recognition. Hoai et al. [8] presented a learning framework that simultaneously performed temporal segmentation and action recognition in time series using dynamic programming and multi-class SVM. Dynamic programming was also used by Shi et al. [23] to segment temporal sequence, but they instead employed semi-Markov models (SMM) for the recognition. Borzehsi et al. [24] proposed extended hidden Markov models with irregular observations (HMM-MIO) which are characterized by high-dimensionality. More recently, Kulkarni et al. [25] designed a dynamic frame warping (DFW) framework based on the template-based representation of action classes and alignment process that assigns a label to each video frame. Contrary to these works, which still require full annotation with pre-segmentation during the training process, our model is able to learn multiple actions simultaneously with weakly annotated data that only indicate the presence of actions.

Action localization: The second category has been widely studied in the vision literature. Several works [14], [26] have attempted to reduce the computational complexity of sliding window technique. Cao et al. [14] performed action detection via efficient pattern search using a branch-and-bound algorithm. Tian et al. [26] generalized deformable part-based model from 2D images to 3D spatiotemporal volumes to detect actions in videos. To further reduce search space of sliding window approaches, recent works [27]–[29], [51], [52] have employed action proposal approaches for locating the potential actions. The work of Jain et al. [27] hierarchically merged super-voxels to generate action proposals called tubelets. Van Gemert et al. [29] generated action proposals directly from dense trajectory features using an efficient clustering algorithm. Yu and Yuan [28] utilized human detector to generate candidate bounding boxes of high actionness scores, and then employed a greedy search method to select the proposals. More recently, deep-learning based approaches have been proposed by Peng et al. [51] and Saha et al. [52], where they employed the supervised proposal networks to extract region proposals. Unlike these methods, we do not use the discriminative model and search strategy to localize the target actions, but instead, exploit the joint-learning of multiple actions within a generative topic model.

C. Topic model for action recognition

Probabilistic topic model has been widely adapted to computer vision tasks, where most studies have addressed the problems of image understanding [31]–[33] and crowded scene analysis [34], [35]. There are several works [36]–[39] applying topic models to solve action recognition problems. Niebles et al. [36] directly applied LDA [10] and pLSA [30] with cuboid features [17] to build recognition models. Rather than using local features, Wang and Mori [37] treated each video frame as a word, and presented the semi-latent LDA model, where latent topics in LDA became observed action labels during training. Wang et al. [39] also used frame-based features but they trained the pLSA model in a supervised manner. Although [37] and [39] obtained the good performance in several standard datasets, both of them required the costly pre-processing stage of human tracking. Bregonzio et al. [38] proposed a supervised topic model to select more informative features for action classification. There are significant differences between our topic model and the others [36]–[39]. First, previous models only produce the two-level representation where the topic is at the highest level and directly relevant to the action class. Instead, the topics in our three-level model are the mid-level representation and shared across high-level actions, allowing us not only bridge the semantic gap but also improve the discriminative power. Meanwhile, these issues can not be addressed with two-level models whose performance is easily limited when dealing with similar actions (e.g., running and jogging). Second, all previous models are parametric and so need to specify the number of topics in advance. In contrast, our topic model is nonparametric and this number is automatically determined from the data. Finally, previous works mainly focus on solving the classification problem, while our proposed model with multi-labeling property enables us to effectively solve various recognition problems, e.g. classification, segmentation, and localization.

III. Proposed Methodology

We now present our proposed methodology in this section and illustrate its framework in Fig. 1. Given a collection of labeled and unlabeled videos, our objective is to jointly learn multiple actions in a single generative model. Using the learned model, we can perform various tasks of action recognition on unlabeled videos. In our proposed model, visual words, often co-occurring in the same video, are composed into atomic actions. Subsequently, co-occurring atomic actions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2816960, IEEE Transactions on Circuits and Systems for Video Technology

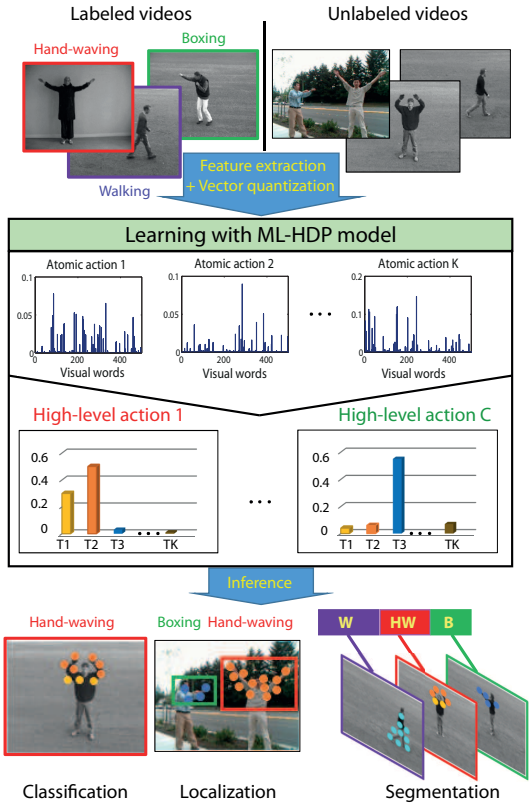


Fig. 1. Overview of proposed framework. We first extract and quantize low-level features to represent visual words. Through our ML-HDP model, mid-level atomic actions (i.e. topics) are modeled as different distributions over visual words while high-level actions are modeled as different distributions over atomic actions. Based on these distributions, each high-level action is related to an appropriate number of topics and the topic will be assigned to each visual word. Topic assignment then helps to perform three recognition tasks (i.e., classification, segmentation, and localization) on unlabeled videos. Here, different colors of circles denote different topics of visual words and the colors of topic assignments in example video are identical to the bar colors of distributions of high-level actions.

are composed into high-level actions. Therefore, our framework connects low-level visual features to atomic actions and high-level actions in a hierarchical structure. In the following subsections, we will describe the details of the framework.

A. Multi-Label Hierarchical Dirichlet Process (ML-HDP)

In topic modeling, LDA [10] is one of the most popular approaches to learning topics from documents. When learning an LDA model, the number of topics need to be specified in advance. However, this can lead to underfitting or overfitting if the setting is unsuitable for the data. In practice, this number is unknown, and it is hard to determine an exact number. To overcome this problem, HDP [11] was proposed as a nonparametric extension of LDA. This model automatically learns the number of topics from data by modeling documents with countably infinite mixture components. In terms of three-level action representation, we consider visual features, atomic actions, and high-level actions as words, topics, and document labels, respectively. Although LDA and HDP with two-layer structure allow us to effectively discover atomic actions from visual features, these unsupervised models do not incorporate

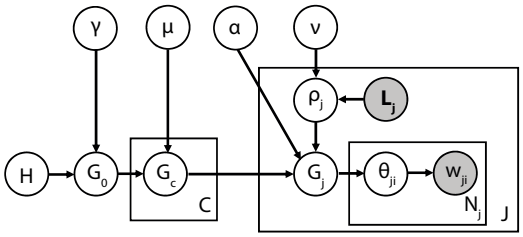


Fig. 2. Graphical model of ML-HDP

the notion of labels and fail to model high-level actions. Hence, they are unable to produce the three-level representation.

We now describe the generative process of our ML-HDP and show it in Fig. 2 (Model parameters are summarised in Table I). In particular, we have extended HDP from two-layer structure to three-layer structure by including one more layer of action classes together with label information. Moreover, ML-HDP is built by applying three stick-breaking constructions [11] successively, first on the corpus, second on the class layer and then on the document layer.

TABLE I
SUMMARY OF MODEL PARAMETERS

γ, μ, α	Concentration parameters
H	Symmetric Dirichlet distribution over vocabulary simplex
$\boldsymbol{\rho_j}$	Mixing proportion of action classes in j^{th} video
$\boldsymbol{\nu}$	Hyper-prior for $\boldsymbol{\rho}$
$\boldsymbol{\beta_0} = \{\beta_{0k}\}$	Topic mixture of entire corpus
$\boldsymbol{\beta_c} = \{\beta_{ck}\}$	Topic mixture of c^{th} class
$\boldsymbol{\pi_j^{(c)}} = \{\pi_{jk}^{(c)}\}$	Topic mixture of c^{th} class in j^{th} video

On the corpus layer, the global measure G_0 is distributed as a Dirichlet process (DP): $G_0|\gamma, H \sim DP(\gamma, H)$. We can express G_0 using a stick-breaking process:

$$G_0 = \sum_{k=1}^{\infty} \beta_{0k}\delta_{\phi_k} \qquad \begin{matrix}\phi_k \sim H \\ \boldsymbol{\beta_0} \sim GEM(\gamma)\end{matrix} \tag{1}$$

Here, $\{\phi_k\}$ are the per-corpus topic distributions over words and δ_{ϕ_k} denotes a Delta function with support point at ϕ_k.

On the second layer, as shown in Fig. 2, high-level actions have random measures $\{G_c\}_{c=1}^{C}$, where C is the number of action classes. Given the base measure G_0, these are drawn from DP: $G_c|\mu, G_0 \sim DP(\mu, G_0)$. As shown in [11], the topic mixture $\boldsymbol{\beta_c} = \{\beta_{ck}\}$ from G_c is also distributed as a DP given the topic mixture $\boldsymbol{\beta_0}$ from G_0. Then, G_c can be expressed by a following stick-breaking process:

$$G_c = \sum_{k=1}^{\infty} \beta_{ck}\delta_{\phi_k} \qquad \boldsymbol{\beta_c} \sim DP(\mu, \boldsymbol{\beta_0}) \tag{2}$$

To address the multi-label problem, we incorporate the notion of multiple actions per video into our model. Particularly, given a collection of J videos, let $\mathbf{w_j} = (w_{j1}, ..., w_{jN_j})$ denote N_j words in video j, where $w_{ji} \in \{1, ..., W\}$ and W is vocabulary size. Let $\mathbf{L_j} = \{c_{jn}\}_{n=1}^{|\mathbf{L_j}|}$ be a set of label indexes, where $c_{jn} \in \{1, ..., C\}$ and $|\mathbf{L_j}|$ corresponds to the number of labels. In our model, both $\mathbf{w_j}$ and $\mathbf{L_j}$ are observed. For each video j, a random measure G_j is drawn from DP with

the base probability measure which is the mixture of random measures $\{G_{c_{jn}}\}_{n=1}^{|\mathbf{L_j}|}$:

$$G_j|\alpha, \{G_{c_{jn}}\}_{n=1}^{|\mathbf{L_j}|} \sim DP\left(\alpha, \sum_{n=1}^{|\mathbf{L_j}|} \rho_{jn} G_{c_{jn}}\right) \tag{3}$$

Here, ρ_{jn} denotes the probability of n^{th} action appeared in video j. The mixing parameter $\boldsymbol{\rho_j}$ is drawn from a symmetric Dirichlet prior $\boldsymbol{\nu}$ given a label set $\mathbf{L_j}$:

$$\boldsymbol{\rho_j} = (\rho_{j1}, ..., \rho_{j|\mathbf{L_j}|}) \sim Dir(\boldsymbol{\nu}|\mathbf{L_j}) = Dir(\nu_1, ..., \nu_{|\mathbf{L_j}|})$$

The third layer of the per-video stick-breaking process for G_j can be derived as follows:

$$\begin{aligned} G_j &= \sum_{s=1}^{\infty} \ddot{\pi}_{js} \delta_{\ddot{\psi}_{js}} & \ddot{\psi}_{js} &\sim \sum_{n=1}^{|\mathbf{L_j}|} \rho_{jn} G_{c_{jn}} \\ & & \ddot{\pi}_{js} &\sim GEM(\alpha) \\ &= \sum_{n=1}^{|\mathbf{L_j}|} \sum_{t=1}^{\infty} \tilde{\pi}_{jt}^{(c_{jn})} \delta_{\psi_{jt}^{(c_{jn})}} & \tilde{\pi}_{jt}^{(c_{jn})} &= \ddot{\pi}_{js} | La(js) = c_{jn} \\ & & \psi_{jt}^{(c_{jn})} &= \ddot{\psi}_{js} | La(js) = c_{jn} \\ &= \sum_{n=1}^{|\mathbf{L_j}|} \sum_{k=1}^{\infty} \pi_{jk}^{(c_{jn})} \delta_{\hat{\phi}_k^{(c_{jn})}} & \pi_{jk}^{(c_{jn})} &= \sum_{t|k_{jt}=k} \tilde{\pi}_{jt}^{(c_{jn})} \end{aligned} \tag{4}$$

Here, $\psi_{jt}^{(c_{jn})}$ and $\pi_{jt}^{(c_{jn})}$ are identical to $\ddot{\psi}_{js}$ and $\ddot{\pi}_{js}$ if $\ddot{\psi}_{js}$ belongs to label c_{jn} ($La(js) = c_{jn}$). Furthermore, since the mixture of random measures is discrete, multiple $\psi_{jt}^{(c_{jn})}$ can be the copies of global parameter $\hat{\phi}_{k_{jt}}^{(c_{jn})}$, where k_{jt} is the index of $\hat{\phi}_k^{(c_{jn})}$ associated with $\psi_{jt}^{(c_{jn})}$, and $\hat{\phi}_k^{(c_{jn})}$ indicates the global parameter ϕ_k from $G_{c_{jn}}$. For each video j, let $\{\theta_{ji}\}_{i=1}^{N_j}$ be i.i.d. random variables distributed as G_j. Then, each visual word w_{ji} is parameterized by a variable θ_{ji} with a multinomial distribution. We have the following likelihoods:

$$\begin{aligned} \theta_{ji}|G_j &\sim G_j \\ w_{ji}|\theta_{ji} &\sim Multi(\theta_{ji}) \end{aligned} \tag{5}$$

Since each factor θ_{ji} is drawn from a discrete random measure G_j, it takes on value ϕ_k associated with label c_{jn} with probability $\pi_{jk}^{(c_{jn})}$. Then, the current topic assignment z_{ji} of word w_{ji} is considered as an indicator variable such that $\theta_{ji} = \phi_{z_{ji}}$. Otherwise, we introduce a variable l_{ji} to indicate the current label assignment of visual words w_{ji}, where $l_{ji} \in \{c_{j1}, ..., c_{j|\mathbf{L_j}|}\}$. Here, l_{ji} and z_{ji} are jointly chosen from the topic mixture $\boldsymbol{\pi_j}$ in video j. Thus, the generative process for ML-HDP is obtained as follows:

$$\begin{aligned} \phi_k|H &\sim H \\ \boldsymbol{\beta_0}|\gamma &\sim GEM(\gamma) \\ \boldsymbol{\beta_c}|\mu, \boldsymbol{\beta_0} &\sim DP(\mu, \boldsymbol{\beta_0}) \\ \boldsymbol{\pi_j} &= (..., \pi_{jk}^{(c_{j1})}, ..., \pi_{jk}^{(c_{j|\mathbf{L_j}|})}, ...) \\ l_{ji}, z_{ji}|\boldsymbol{\pi_j} &\sim Multi(\boldsymbol{\pi_j}) \\ w_{ji}|z_{ji}, \{\phi_k\}_{k=1}^{\infty} &\sim Multi(\phi_{z_{ji}}) \end{aligned} \tag{6}$$

As shown in the generative process of Eq. 6, the topic mixtures of classes $\{\boldsymbol{\beta_c}\}$ are generated from the topic mixture of corpus $\boldsymbol{\beta_0}$. This results in different classes are associated with different mixtures over topics. These mixtures are considered as model parameters used to generate the topic mixture $\boldsymbol{\pi_j}$ in video j. Then, videos containing the same classes would share similar atomic actions, since each high-level action is associated with an appropriate number of atomic actions. Subsequently, each visual words w will be chosen with the probability that measures how likely an action l and an atomic action z co-appear in the video, and how likely the word w is related to that atomic action. Thus, the generative process exhibits the property of three-level action representation.

It is worth noting that although our model is relevant to the HDP model proposed in [34] in terms of three-level structure, there are three major differences between two models. First, HDP [34] is unsupervised and particularly designed for clustering video clips. In contrast, by incorporating label information (e.g., $\mathbf{L_j}$), ML-HDP acts more discriminatively and better learn which atomic actions occur in each high-level action. Second, HDP [34] is limited to involve each video with only one cluster label, which is equivalent to high-level action in our work, and it is unable to model videos with multiple labels. More specifically, G_j of a video in [34] is drawn from DP with one of base probability measures $\{G_c\}$, whereas G_j in ML-HDP is distributed with the mixture of random measures $\{G_c\}$ (see Eq. 3) to explicitly model multi-labeled videos. Third, in [34], each visual word w is only associated with topic z and not directly connected to high-level classes. However, as shown in the generative process of ML-HDP (see Eq. 6), visual word w is associated with both topic z and label l. This not only captures more distinctive information but also allows us to handle challenging scenarios like multiple actions present in a video frame. To validate these points, we will evaluate our model with real-world datasets in the experiment section. Thus, our model is more general and better deal with practical recognition problems.

B. Posterior perspective of ML-HDP

In this section, we describe a posterior perspective for ML-HDP. Together with stick-breaking process, this perspective is essential to the development of inference algorithm. Following a Pólya urn scheme [11], the draws from DP are discrete and posses a clustering property. As a result, the mixture components $\{\phi_k\}$ in this perspective are shared across multiple DPs. Let factor $\tilde{\phi}_{cd}$ be the draw from G_0. As summarized in Table II, we define counting notations n_{jck}, m_{jck}, and q_{ck} to represent associations between these factors with mixture components $\{\phi_k\}$. Then, we derive the construction for G_0, $\{G_c\}$ and $\{G_j\}$ using the posterior structure as in [12].

TABLE II
SUMMARY OF COUNTING NOTATIONS

n_{jck}	No. of $\{\theta_{ji}\}$ relevant to $\hat{\phi}_k^{(c)}$ in document j (refer to Eq. 4)
m_{jck}	No. of $\{\psi_{jt}^{(c)}\}$ in document j relevant to ϕ_k (refer to Eq. 2)
q_{ck}	No. of $\{\tilde{\phi}_{cd}\}$ in class c is relevant to ϕ_k (refer to Eq. 1)

First, similar to [11], we represent the marginal counts with dots. Let $q_{.k}$ be the number of $\{\tilde{\phi}_{cd}\}$ in all classes associated with ϕ_k. A construction for G_0 can be expressed as follows:

$$(\beta_{0u}, \beta_{01}, ..., \beta_{0K}) \,|\gamma, G_0, \tilde{\boldsymbol{\phi}} \sim Dir(\gamma, q_{.1}, ..., q_{.K}) \tag{7}$$

$$G_{0u}|\gamma, H \sim DP(\gamma, H) \tag{8}$$

$$G_0 = \sum_{k=1}^{K} \beta_{0k}\delta_{\phi_k} + \beta_{0u}G_{0u} \tag{9}$$

Second, let $m_{.ck}$ be the number of $\{\psi_{jt}^{(c)}\}$ in the corpus associated with ϕ_k. Then, G_c can be constructed as follows:

$$(\beta_{cu}, \beta_{c1}, ..., \beta_{cK})|\mu, \boldsymbol{\psi^{(c)}} \sim Dir(\mu\beta_{0u}, m_{.c1} + \mu\beta_{01}, ..., m_{.cK} + \mu\beta_{0K}) \tag{10}$$

$$G_{cu}|\mu, G_0 \sim DP(\mu\beta_{0u}, G_{0u}) \tag{11}$$

$$G_c = \sum_{k=1}^{K} \beta_{ck}\delta_{\phi_k} + \beta_{cu}G_{cu} \tag{12}$$

Third, a construction for G_j is now given as:

$$(..., \pi_{ju}^{(c_{jn})}, ..., \pi_{jk}^{(c_{jn})}, ...)|\alpha, \boldsymbol{\rho_j}, \boldsymbol{\theta_j} \sim Dir(..., \alpha\rho_{jn}\beta_{c_{jn}u}, ..., n_{jc_{jn}k} + \alpha\rho_{jn}\beta_{c_{jn}k}, ...) \tag{13}$$

$$G_{ju}|\alpha, \{G_{c_{jn}}\}, \boldsymbol{\rho_j} \sim DP(\alpha \sum_n \rho_{jn}\beta_{c_{jn}u}, \sum_n \rho_{jn}G_{c_{jn}u}) \tag{14}$$

$$G_j = \sum_{n=1}^{|\mathbf{L_j}|}\sum_{k=1}^{K} \pi_{jk}^{(c_{jn})}\delta_{\hat{\phi}_k^{(c_{jn})}} + \sum_{n=1}^{|\mathbf{L_j}|} \pi_{ju}^{(c_{jn})}G_{ju} \tag{15}$$

Eqs. 9, 12, and 15 show that the posterior for G_0, G_c, and G_j contains two parts. The first part show that posteriors share the same set of topics $\{\phi_k\}$, but the mixtures over topics are different. The second part (i.e. G_{0u}, G_{cu}, G_{ju}) are distributed as DP, which have the same structure as original ML-HDP, but concentration parameters are changed.

C. Inference in the ML-HDP model

In this section, we will describe how to infer model parameters and latent variables from data. However, exact inference in DP based models is intractable. Instead, we use a collapsed Gibbs sampling algorithm [40] for approximate inference. This algorithm is performed by: (1) sampling latent variables assuming that model parameters are given; (2) sampling model parameters assuming that latent variables are given. This process will be iterated until convergence.

In our model, the Gibbs sampling algorithm using direct assignment [11] is derived from the posterior process described in Section III-B. In particular, referring to Eqs. 9, 12, and 15, the second part of these equations (i.e. $G_0u, \{G_{cu}\}, \{G_{ju}\}$) and the per-document topic mixture $\{\boldsymbol{\pi_j}\}$ are integrated out without sampling. In addition, since H is conjugate to the multinomial likelihood for the visual words, we also integrate out the mixture components $\{\phi_k\}$ during the sampling schemes. Thus, the Gibbs sampler maintains: (1) the latent variables including topic assignments $\mathbf{z}$ and label assignments $\mathbf{l}$; and (2) the model parameters including the per-corpus topic mixture $\boldsymbol{\beta_0}$ and per-class topic mixture $\boldsymbol{\beta_c}$. Otherwise, a set of auxiliary variables related to the counts of model factors (i.e. $\mathbf{m}, \mathbf{q}$) are also needed to sample. The procedure of Gibbs sampling is described as follows:

Sampling z and l. We consider the joint probability of the topics and labels since these two variables are dependent (see Eq. 6). Then, by integrating out parameters $\pi_{\mathbf{j}}$ and $\{\phi_k\}$, the update formula that allows joint sampling topic assignment z_{ji} and label assignment l_{ji} is given by:

$$p(z_{ji} = k, l_{ji} = c_{jn}|\mathbf{z^{-ji}}, \mathbf{l^{-ji}}, \{\boldsymbol{\beta}_{\mathbf{c_{jn}}}\}, \alpha, \boldsymbol{\rho_j}) \propto \begin{cases} \left(n_{jc_{jn}k}^{-ji} + \alpha\rho_{jn}^{(-ji)}\beta_{c_{jn}k}\right) f_{c_{jn}k}^{-w_{ji}}(w_{ji}) & \text{if } k \text{ existing} \\ \alpha\rho_{jn}^{(-ji)}\beta_{c_{jn}u} f_{c_{jn}k_{new}}^{-w_{ji}}(w_{ji}) & \text{if } k = k_{new} \end{cases} \tag{16}$$

Here, $f_{ck}(w_{ji})$ is the conditional density of w_{ji} under mixture component w from class c given all data items except w_{ji}. Let $\mathbf{w_{ck}} = \{w_{ji}|z_{ji} = k, l_{ji} = c, \text{ all } j, i\}$. We have:

$$f_{ck}^{w_{-ji}}(w_{ji}) = \frac{\int f(w_{ji}|\phi_k) \prod_{\substack{j'i' \neq ji \\ w_{j'i'} \in \mathbf{w_{ck}}}} f(w_{j'i'}|\phi_k)h(\phi_k)d\phi_k}{\int \prod_{\substack{j'i' \neq ji \\ w_{j'i'} \in \mathbf{w_{ck}}} f(w_{j'i'}|\phi_k)h(\phi_k)d\phi_k} \tag{17}$$

If the new topic is chosen ($k = k_{new}$), then the weight of new component is instantiated in the sampler. $f_{ck_{new}}^{w_{-ji}}(w_{ji}) = \int f(w_{ji}|\phi_k)h(\phi_k)d\phi_k$ is the prior density of w_{ji}.

Sampling m. For each document j, the auxiliary variable m_{jck} is sampled as:

$$p(m_{jc_{jn}k} = m|\mathbf{z}, \mathbf{l}, \mathbf{m_{-jc_{jn}k}}, \{\beta_{c_{jn}}\}, \alpha, \boldsymbol{\rho_j}) \propto s(n_{jc_{jn}k}, m)\left(\alpha\rho_{jn}\beta_{c_{jn}k}\right)^m \tag{18}$$

where $s(.,.)$ are unsigned Stirling numbers of the first kind, which is defined as: $s(0,0) = s(1,1) = 1; s(n,0) = 0; s(n,m) = 0$ for $m > n; s(n+1,m) = s(n,m-1) + ns(n,m)$.

Sampling $\boldsymbol{\beta_c}$. For each class c, we first accumulate m_{jck} for all documents j's to get $(m_{.c1}, ..., m_{.cK})$. Then, $\boldsymbol{\beta_c}$ is sampled using Eq. 10.

Sampling q. For each class c, the auxiliary variable q_{ck} is sampled as:

$$p(q_{ck} = q|\mathbf{m}, \mathbf{q_{-ck}}, \mu, \boldsymbol{\beta_0}) \propto s(m_{.ck}, q)\,(\mu\beta_{0k})^q \tag{19}$$

Sampling $\boldsymbol{\beta_0}$. Similar to sampling $\boldsymbol{\beta_c}$, we accumulate q_{ck} for all classes c's to get $(q_{.1}, ..., q_{.K})$. Then, we sample $\boldsymbol{\beta_0}$ using Eq. 7.

Our Gibbs sampling algorithm is further summarized in term of pseudocode as shown in Algorithm 1. In particular, to initialize, we randomly assign values of to $\mathbf{z}$ and $\mathbf{l}$. According to the initial assignment, auxiliary variables $\mathbf{m}$ are sampled using Eq. 18. Then, model parameters $\boldsymbol{\beta_c}$ are initialized from sampling Eq. 10. Similarly, $\boldsymbol{\beta_0}$ are initialized using Eq. 7 via auxiliary variables $\mathbf{q}$

Algorithm 1 Gibbs sampling for ML-HDP

Input: Corpus of video data formed as a bag of visual words $\mathbf{w_j} = (w_{j1}, ..., w_{jN_j})$ and a set of label indexes $\mathbf{L_j} = \{c_{jn}\}_{n=1}^{|\mathbf{L_j}|}$.

Output: Latent variables $\mathbf{z}$, $\mathbf{l}$, and model parameters $\boldsymbol{\beta_c}$, $\boldsymbol{\beta_0}$

1) **Initialization**. Initialize latent variables and a set of parameters $\{\boldsymbol{\beta_c^{(0)}}, \boldsymbol{\beta_0^{(0)}}\}$
2) **For each** $it = 1, ..., M$ **do**:
   a) **For each** $j = 1, ..., J$ **do**:
      i) **For each** $i = 1, ..., N_j$ **do**:
         Given $\{\boldsymbol{\beta_c^{(it-1)}}\}$, jointly sample z_{ji} and l_{ji} using Eq. 16.
      ii) **For each** $(c_{jn}, k) \in \{c_{jn}, k | n = 1...|\mathbf{L_j}|, k = 1...K\}$ **do**:
         Sample $m_{jc_{jn}k}$ using Eq. 18
   b) **For each** $c = 1, ..., C$ **do**:
      i) Given $\{\boldsymbol{\beta_0^{(it-1)}}\}$, sample $\boldsymbol{\beta_c}$ using Eq. 10
      ii) **For each** $k = 1, ..., K$ **do**:
         Sample q_{ck} using Eq. 19
   c) Sample $\boldsymbol{\beta_0}$ using Eq. 7
3) **End**

D. Action recognition

For action recognition, we divide a collection of videos into train and test sets, where the labels of train videos are observed while the labels of test videos are unseen. Then, our recognition framework can be applied in a semi-supervised fashion, where we predict the labels of videos in test set from the observed labels of train set. Here, we assume that each test video is associated with all C possible labels. ML-HDP is run on the entire collection to infer the latent variables using the Gibbs sampling algorithm. Since each visual word is labeled as one of C actions after sampling, this allows us to develop straightforward methods for action classification as well as multiple action recognition.

1) Action classification: In this task, we classify the test video j into an action label having the most number of words assigned. Formally, action label of video j is predicted as following:

$$c_j^* = \underset{c}{\operatorname{argmax}}\, n_{jc.} = \underset{c}{\operatorname{argmax}} \sum_k n_{jck} \tag{20}$$

Here, the marginal count $n_{jc.}$ (refer to Table II) is the number of words in video j assigned to label c.

Besides above method, once the ML-HDP model is learned, it can be applied to encode local features by representing a video j as a topic proportion vector $\mathcal{V}_j = [n_{j.1}/n_{j..}...n_{j.K}/n_{j..}]$. The encoded vectors are then used to train and test SVM classifier. We will further examine the encoding ability of ML-HDP in our experiments.

2) Multiple action recognition: Due to the nature of bag-of-words representation, topic model is not aware of relative temporal ordering of features, which is essential to assign labels at different time instances. Hence, following [25], we perform multi-action recognition based on the per-frame representation, where each frame is also represented as a bag-of-words like video representation. Specifically, we adopt an adaptive-size window Δ_f around each frame f as presented in [25] to avoid frames containing no features. The adaptive-size window is created by symmetrically growing the window size until accumulating the predefined number Q of features. Compared to the fixed-size window, this type of representation helps us to better address the issues of variabilities in the speed of different subjects. Subsequently, given the window with a set of labeled words, we perform various types of multi-action recognition including continuous action segmentation and action localization.

For action segmentation, each video frame f contains only one action corresponding to the label with the most number of words assigned. Formally, we can predict the action label of a frame f in video j as below:

$$c_j^{(f)} = \underset{c}{\operatorname{argmax}}\, N_{jc}^{(f)} = \underset{c}{\operatorname{argmax}} \sum_i n_{jc.}^{(i)} \tag{21}$$

Where, $f - 0.5\Delta_f \leq i \leq f + 0.5\Delta_f$; $n_{jc.}^{(i)}$ denotes the number of words at frame i assigned to c; and $N_{jc}^{(f)}$ denotes the number of words within window Δ_f assigned to c.

For action localization, within the window Δ_f, we only keep the significant actions whose number of assigned words is greater than a predefined threshold Th. Moreover, since each frame is classified into one or more labels, the segment of a target action can be localized in the temporal dimension. Here, each video is composed of multiple segments, those can be overlapped in some frames. Then, an action c of the spatiotemporal segment in video j is scored as below:

$$R_{jc} = \frac{1}{F_e - F_s + 1} \sum_{f=F_s}^{F_e} \frac{N_{jc}^{(f)}}{N_{j.}^{(f)}} \tag{22}$$

Where, $N_{jc}^{(f)} \geq Th$; F_s and F_e denote start and end frames of the segment; and $N_{j.}^{(f)}$ denotes the total number of words within window Δ_f.

Discussion. Based on the bag-of-words paradigm, while ML-HDP is developed for recognizing actions in videos, it is also applicable to other vision tasks in image domain such as object recognition or scene understanding. This is done by simply replacing local video features with local image features, and latent topics can be viewed as objects or object parts. However, local features extracted from image patches are usually not discriminative enough across scenes or object classes [37]. Accordingly, to achieve the good performance, it is essential to integrate the geometric relation of image features into recognition models as revealed in previous works [16], [31]. Meanwhile, using local space-time features, conventional topic models like LDA [37] and pLSA [36], which neglect the geometric arrangement of these features, have been proven effective to capture well the co-occurring spatiotemporal patterns. This is because the additional discriminative cues of video features are provided by human motions. Motivated by conventional models, we do not incorporate the structural information among visual features to avoid increasing the

model complexity. Instead, we are interested in investigating the versatility of the generative topic model and the robustness of three-level representation for different tasks in action recognition. Therefore, within the scope of this paper, we prefer to use ML-HDP for video domain rather than for image domain.

IV. Experiments

In this section, we describe the detailed empirical study of our proposed approach. We first introduce the experimental settings and datasets used for our evaluation. We then extensively present the results obtained for topic visualization, action classification, and multi-action recognition.

A. Datasets and experimental settings

We report experimental results on four publicly available datasets: KTH [13], MSR-II [14], Hollywood2 [55], and UCF101 [56]. The description of each dataset is as following:

- **KTH**: This dataset contains 598 video sequences of 25 subjects. Each subject performs 6 action classes. Video sequences are taken over homogeneous background with static camera. For action classification, we follow the standard setting as in [13]. The performance is measured by the average accuracy. For continuous action segmentation, we use a stitched version of KTH, namely s-KTH. Following [24], the stitched dataset is built by simply concatenating individual action instances into sequences. Totally, there are 64 and 36 multi-action sequences used for training and testing, respectively. Similar to [8], [24], the performance of continuous action segmentation is measured via frame-wise accuracy. Concretely, since each frame is associated with a label, the overall frame-wise accuracy is the percentage of true positives.
- **MSR-II:** The MSR-II dataset is particularly used for the evaluation of action localization, where it contains three action classes selected from KTH: boxing, hand-clapping, and hand-waving. MSR-II consists of 203 actions in 54 videos. Actions are performed in a crowded environment with cluttered background. Different from s-KTH, some videos in MSR-II contain multiple actions even occurring at the same time, which make this dataset more realistic and challenging. As in [14], the performance is measured by computing the average precision (AP) score.
- **Hollywood2:** This dataset contains 1,707 videos with 12 action classes. For action classification, the performance is measured by mean average precision (mAP). For action segmentation, we use the subset of Hollywood2 called HOHA [2], which contains 430 videos with 8 action classes. Since each video sample contains a single action, following [8], we concatenate eight randomly selected original video samples to generate 30 long testing sequences. We also use the frame-wise accuracy like s-KTH. Generally, Hollywood2 is very challenging because each video sequence contains significant camera motion, rapid scene changes, and occasionally significant clutter.
- **UCF101:** This is one of largest and most challenging datasets that consists of 13,320 video clips with 101 action classes collected from Youtube in realistic scenarios. For action classification, we conduct evaluation according to three train/test splits as in [56] and report the mean average accuracy of these splits. The subset of this dataset, which has 3,204 videos with 24 classes, is used for the evaluation of action localization. We use the first split of train and test sets with 2,290 and 914 videos, respectively. The performance is measured by mAP.

Feature extraction: To verify the compatibility of ML-HDP, we perform experiments with three types of local features including two hand-crafted features (i.e. STIP [1] and IDT [3]) and one deep-learned feature [50]. These low-level features have shown the excellent performance on a variety of datasets. Each feature is described as below:

- **STIP.** This type of feature utilizes 3D-Harris detector to find a set of sparse interest points capturing the regions of high motion salience. The volume around each interest point is then described by concatenating HOG and HOF descriptors to form a 162-dimensional vector.
- **IDT.** This is the improved version of Dense Trajectories (DT) [4], where the feature points in each frame are densely sampled and tracked using optical flow. To improve the performance of DT, IDT stabilizes optical flow to eliminate camera motion. Different kinds of descriptors are then computed along the trajectories to capture shape, appearance, and motion information such as HOG, HOF, and MBH. Here, we use MBH descriptor (192 dimension) due to its best performance compared to HOG and HOF.
- **sTDD.** Trajectory-pooled deep-convolutional descriptor (TDD) [50] is a deep-learned variant of IDT, which is computed by pooling CNN feature maps along trajectories. In experiments, we leverage the filters of Conv4 and Conv5 layers from VGG16 CNN model [57] as extractors. We only employ the spatial net and do not use the temporal net [50] with optical flow images to reduce model complexity and storage requirements. Then, we denote this feature as sTDD. To de-correlate and reduce high-dimensional features, we apply PCA with whitening. The dimension of sTDD is reduced from 1024 to 128. Unlike [50], which only tracks sampled points on a single scale, we track sampled points on multiple scales as the original implementation of IDT.

Codebook generation: For each feature type, we randomly sample 500,000 descriptors from subsets of action datasets to perform k-means and quantize a set of local features into visual words. Empirically, for densely sampled features (i.e. IDT and sTDD), we select the codebook sizes of 2,000, 4,000, and 10,000 for KTH, Hollywood2, and UCF101, respectively. For the sparse feature STIP, the codebook sizes are 500, 3,000, and 5,000 for KTH, Hollywood2, and UCF101 respectively.

B. Topic learning and visualization

In this section, we visualize topics discovered by our ML-HDP model during learning process on the KTH dataset. The topic learning of our model is achieved by running the collapsed Gibbs sampling until convergence. We also interpret the use of discovered topics for action recognition.

As described in Section III, through ML-HDP, the topics help us to explain typical atomic actions in the scene. A high-level action c is interpreted by the topic mixtures $\boldsymbol{\beta_c}$. Figures 3(a) and 4(a) depict the parameters $\{\boldsymbol{\beta_c}\}$ of 6 action classes estimated from KTH using IDT and STIP features respectively (Note: since IDT and sTDD are two versions of the trajectory-based feature, we only examine the topic learning of IDT in this experiment). ML-HDP automatically discovers 31 topics with IDT features, and 23 topics with STIP features. We can see that each high-level action has a different weighted combination of topics. Particularly, in Table III, we summarize the set of significant topics for each class. The results of Table III correctly reflect the property of our model that each class tends to be associated with the different set of topics. Hence, extracted topics provide the useful information to identify which actions are present in a video. The pairs of example videos containing the same dominant topic are shown in Figures 3(b) and 4(b) relevant to IDT and STIP features respectively. A topic shown in each example video also corresponds to the one having the highest weight in the topic mixture of action belonging to that video.

TABLE III
EXAMPLE OF EXTRACTED TOPICS FROM KTH DATASET

Action class	IDT feature	STIP feature
boxing	4,14,16,22,24	5, 6, 12, 16, 19, 23
hand-clapping	2,6,13,16,21	3, 13, 18, 22
hand-waving	3,7,25	4,14
jogging	1,17,19,26,31	2, 8, 11, 15, 17, 20
running	19,26,30,31	2, 8, 11, 15, 20
walking	5, 9, 18, 20, 29	1, 7, 8, 9, 10, 17

Interestingly, as shown in Figures 3(a) and 4(a), the topic mixtures $\{\boldsymbol{\beta_c}\}$ of similar actions (i.e., jogging and running) share the common topics with different weights. For example, the significant topics $\{19, 26, 31\}$ are shared between the jogging and running for the IDT feature. While topic 31 is the most significant topic of jogging, topic 30 is the most significant one of running. Figure 5 further illustrates the potential of ML-HDP to differentiate similar actions via the parameters $\{\boldsymbol{\pi_j}\}$, which are generated from $\{\boldsymbol{\beta_c}\}$ (described in Section III-A). Similar observation is drawn with STIP features. On the other hand, the topic mixtures of actions having high inter-class variations also share some common topics, but their weights are very small. Moreover, different topics can be explored in the same action, where the difference between these topics corresponds to the intra-class variability. Thus, through estimated parameters $\{\boldsymbol{\beta_c}\}$, we can see that atomic actions are shared across high-level actions and act as mid-level representations. This allows our ML-HDP model works discriminatively because it not only successfully extracts the co-occurring patterns of visual cues but also captures well the intra-class and inter-class variations.

C. Single action recognition

In this section, ML-HDP is benchmarked for action classification on pre-segmented videos. The first experiment investigates the performance of different bag-of-words methods. The second experiment compares ML-HDP to the state-of-the-art approaches. Finally, we show the comparison of computational cost between ML-HDP and related works.

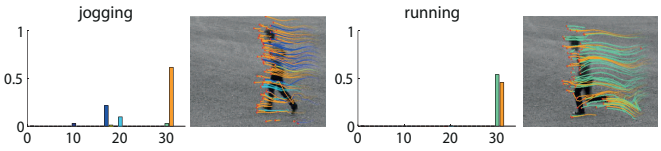


Fig. 5. Illustration of topic mixtures $\boldsymbol{\pi_j}$ in two example videos that capture similar actions performed by a same person. Here, "jogging" and "running" videos share a common topic 31 but their topic mixtures are different.

1) Comparison between bag-of-words methods : This experiment presents the performance of ML-HDP and competitors based on bag-of-words representation, those are categorized into feature encoding and topic modeling techniques. Here, we implemented all these competitors by ourselves. For the fair comparison, we use IDT with MBH descriptor for all methods. The details are specified as following:

- **Feature encoding.** We conduct experiments against popular encoding techniques including: bag of words (BoW), VLAD, and Fisher Vector(FV). For BoW, a video is represented as the histogram over learned codebook. VLAD and FV are the advanced variants of BoW with high-order statistics. Here, the codebook size (VLAD) and the mixture number (FV) are set to 256. Then, the classification is done with linear SVM using one-vs-all training scheme.
- **Topic modeling.** We first employ two supervised (parametric and non-parametric) models for comparison. For the parametric model, we use MedLDA [58]. For the non-parametric model, we modify the unsupervised HDP model proposed in [34] to work under semi-supervised setting, where action labels of training set are observed and we only update the action label for testing set in each sampling iteration. By utilizing label information, we not only improve the distinctive capacity of HDP but also speed up its learning time. Then, we refer this method as semi-HDP. Besides, we also study the encoding ability of topic models. We apply LDA, MedLDA, and semi-HDP to infer the topic proportions, these are used as feature vectors to train and test with linear SVM.

As shown in Table IV, we achieve the highest accuracy with ML-HDP. Not surprisingly, ML-HDP outperforms unsupervised LDA by a large margin. Moreover, ML-HDP associated with feature-level labels is more discriminative than two supervised models associated with document-level labels. In all cases, non-parametric models obtain better results than parametric model. This can be explained that non-parametric models possess the stronger learning ability by automatically discovering the number of latent topics from data. Meanwhile, the parametric MedLDA is very sensitive to the number of topics, especially when dealing with the large-scale and complex dataset like UCF101. Furthermore, by exploiting the higher-level action representation, ML-HDP outperforms popular feature encoding methods, those solely rely on low-level features. Table IV also shows the good encoding ability

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2816960, IEEE Transactions on Circuits and Systems for Video Technology

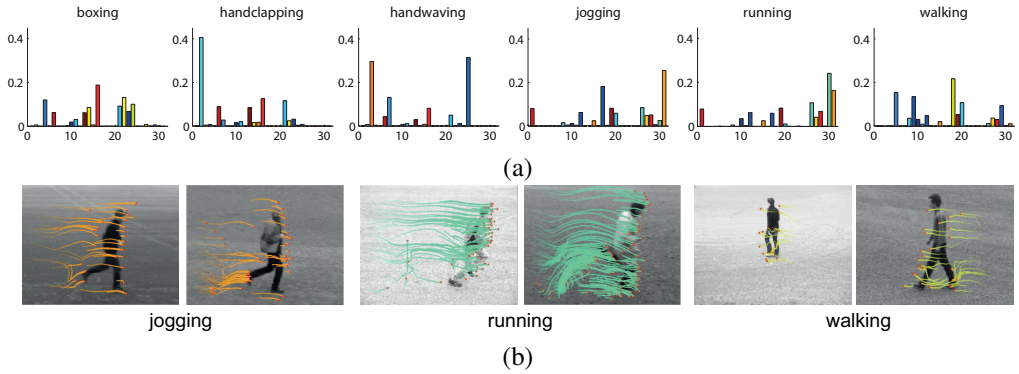


Fig. 3. Illustration of learned topics/high-level actions using IDT features. (a) The mixtures $\{\boldsymbol{\beta_c}\}$ of six action classes over 31 topics. $x-$axis is the index of topics. $y-$axis is the mixture over topics. (b) Pair of example videos with the same dominant topic for three example actions. The colors of topics shown in example videos are identical to the colors of topic indexes in the topic mixtures.

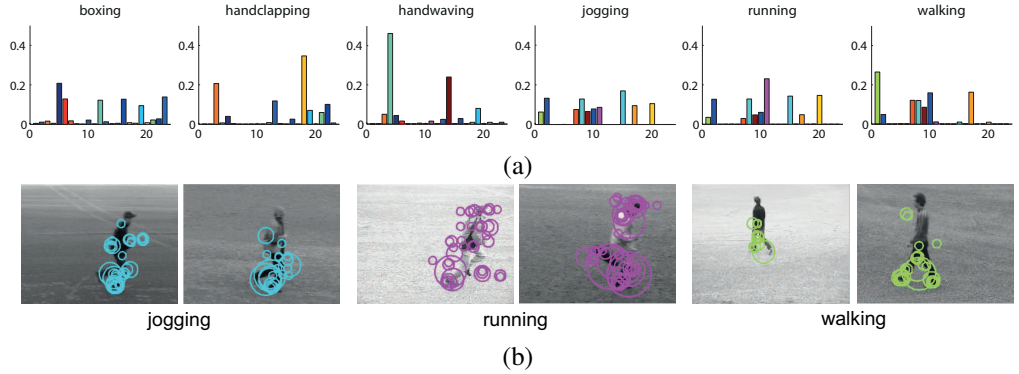


Fig. 4. Illustration of learned topics/high-level actions using STIP features. (a) The mixtures $\{\boldsymbol{\beta_c}\}$ of six action classes over 23 topics. $x-$axis is the index of topics. $y-$axis is the mixture over topics. (b) Pair of example videos with the same dominant topic for three example actions.

of topic models when using topic representation to learn SVM classifier. Compared to pure MedLDA and semi-HDP, the performance of MedLDA^{svm} and semi-HDP^{svm} are improved and comparable to VLAD and FV. ML-HDP^{svm} is even superior to encoding competitors. However, the combination of ML-HDP with SVM does not yield better results because the pure ML-HDP model itself succeeded in capturing co-occurrence patterns of action labels and local features. It should be noted that encoding with topic models is more memory efficient than popular encoding. On large-scale dataset, to achieve reliable performance, the topic proportion vector has a few hundred dimensions (relevant to the number of learned topics), whereas the vector encoded by VLAD or FV has thousands of dimensions. Hence, for applications with memory constraint, topic modeling is the better choice.

2) Comparison with the state of the arts: This experiment reports our results using different features and compares with the state-of-the-art methods, those are divided into four groups as following: models based on low-level features, models based on mid-level representation, models built in a hierarchical manner, and deep-learned models.

We first evaluate the performance of ML-HDP in terms of different features. As shown in Table V, IDT and sTDD are significantly better than STIP on three datasets, suggesting that our model captures motion patterns of dense features better than the ones of sparse features. Note that the number of STIP features are substantially ten times lower than the number of IDT based features. As a result, the convergence of Gibbs sampling with STIP is significantly faster. Hence, there is a trade-off between the processing time and accuracy in the use of local features. For the trajectory-based feature, the deep-learned sTDD descriptor is slightly worse than the hand-crafted MBH descriptor on small-scale KTH, but it performs better on large-scale Hollywood2 and UCF101 by around 2%. Furthermore, by using the same trajectories, we can combine IDT and sTDD at descriptor-level [53] to boost the performance. Since the number of combined features is the same with the number of pure features, the sampling complexity is not changed. Table V demonstrates that the combined feature significantly improves the accuracy, and two features are strongly complementary to each other. While sTDD with spatial net captures the appearance information, IDT with MBH descriptor represents the motion information.

Secondly, from Table V, ML-HDP outperforms most methods on three datasets. Specifically, all low-level feature based models yield inferior performance. For example, we achieve better results than recent advances of feature encoding, those are augmented by spatial Fisher Vector (SFV) and spatial-temporal temporal pyramid (STP) strategies [3]; or stacking features at multiple frame rates (MIFS [42]). Moreover, superior results on Hollywood2 and UCF101 shows that our learned atomic actions are more discriminative than other types of mid-level features. On small-scale KTH, action bank [46] and multi-instance learning [45] perform slightly better

TABLE IV
COMPARISON OF OUR MODEL WITH BAG-OF-WORDS METHODS (SUPERSCRIPT SVM DENOTES TOPIC MODELS COMBINED WITH SVM CLASSIFIERS)

Datasets	BoW	VLAD	FV	LDA^{svm}	MedLDA^{svm}	semi-HDP^{svm}	ML-HDP^{svm}	MedLDA	semi-HDP	ML-HDP
KTH	92.7%	94.2%	94.4%	85.1%	92.4%	93.3%	95.5%	91.9%	93%	**95.8%**
Hollywood2	54.2%	58.1%	58.8%	48.3%	59.1%	61.2%	63.2%	57.7%	60.7%	**63.5%**
UCF101	69.3%	79.2%	80.7%	60.6%	71.5%	79.3%	82.2%	68.1%	78.1%	**83.4%**

than ML-HDP, but their performance is greatly degraded when dealing with large-scale data. In addition, ML-HDP outperforms hierarchical models except for VLAD^3 [47], which utilizes complex 3D CNN features and achieves better accuracy on UCF101. Particularly, we compare ML-HDP with remarkable approaches of hierarchical feature encoding under the same three-level representation. These include FV+StackedFV [60] and its variant (MLFVE [61]) improved by considering geometric relationships among local features. For FV+StackedFV, we follow the setting in Section IV-C1 and the implementation protocol of [60] to obtain results on KTH and Hollywood2, which were not reported in the original paper. Obviously, using multiple nested Fisher layers, FV+StackedFV outperforms conventional encoding approaches, whose results are shown in Table IV. Meanwhile, using the same IDT feature, ML-HDP shows the slight improvement over FV+StackedFV, since our model with incorporated labels provides more distinctive clues. Moreover, ML-HDP with the combined feature yields the better result against MLFVE, which further demonstrates its effectiveness. Finally, our method is on par with results of deep learning methods. Best results of Hollywood2 and UCF101 are 76.7% [48] and 91.5% [50], those deploy highly sophisticated CNN models. Otherwise, most of the best results (e.g., [47], [48], and [50]) from the literature are obtained by combining with hand-crafted features (IDT). Different from our combining strategy, they combine features at video-level leading to very high-dimensional representation and hence low memory-efficiency.

3) Computational cost comparison: The experiments are conducted on a machine with 3.4 GHz CPUs and 32 GB of RAM. Here, we compare with the computational cost of the popular feature encoding and the mid-level feature learning, which is closely related to our topic learning. All methods use trajectory features to report their execution time per video measured from the input of a raw video to the output of the learned or encoded feature. Following [43], results in Table VI are reported on the small-scale UT-I dataset [54]. Not surprisingly, without exploiting higher-level action representation, feature encoding methods run significantly faster than learning-based methods. Action bank and EXMOVES are computationally expensive to extract mid-level features, since they take much time to matching templates [46] or sliding action-exemplars [43] in each video. These methods are almost ten to hundred times slower than ML-HDP. For MIL-F32 [45], we convert their total timing into the average time per video. Note that this result might not be directly comparable to other methods in this experiment, since it was reported on a different dataset. However, MIL-F32 is costly due to the need to encode a large number of action parts (300 to 3,000 FVs per video). Among topic models using Gibbs samplers, the runtime of ML-HDP is slower than the ones of LDA and semi-HDP. The reason is that, in each sampling iteration, ML-HDP jointly samples topic and label assignments (see Eq. 16), while LDA and semi-HDP only sample topic assignment in their update equations. Otherwise, variational inference of MedLDA is faster than Gibbs sampling inference. In general, ML-HDP exhibits the reasonable efficiency in comparison with related works.

TABLE V
COMPARISON WITH THE STATE OF THE ART METHODS ON KTH, HOLLYWOOD2, AND UCF101 FOR ACTION CLASSIFICATION

Methods	**KTH**	**Hollywood2**	**UCF101**
Low-level feature based models			
STP+SFV(IDT) [3]	-	66.8%	86%
MIFS [42]	-	68%	89.1%
Gilbert et al. [41]	94.5%	50.9%	-
BoW(STIP) [5]	91.8%	47.7%	-
Mid-level feature based models			
EXMOVES [43]	-	56.6%	71.6%
MIL-F32 [45]	96.8%	51.7%	-
Action bank [46]	**98%**	-	-
Hierarchical models			
VLAD^3 [47]	-	-	**92.2%**
FV+StackedFV(IDT) [60]	95.4%	63.3%	-
MLFVE [61]	-	67.4%	-
LHM [21]	91.4%	59.9%	-
CNN-based models			
HRP(CNN) + RP(IDT) [48]	-	**76.7%**	91.4%
Jain et al. [49]	95.4%	66.4%	88.5%
FV(IDT) + FV(TDD) [50]	-	-	91.5%
ML-HDP(STIP)	91.3%	54.1%	68.3%
ML-HDP(IDT)	95.8%	63.5%	83.4%
ML-HDP(sTDD)	94.1%	65.7%	85.1%
ML-HDP(IDT+sTDD)	**96.5%**	**68.8%**	**89.3%**

TABLE VI
EXECUTION TIME (IN SECONDS PER VIDEO) OF DIFFERENT METHODS

Low-level feature encoding			**Mid-level feature learning**		
BoW	VLAD	FV	Action Bank	EXMOVES	MIL-F32
72.8	47.9	54	29,700	2580	227

Topic learning			
LDA	semi-HDP	MedLDA	ML-HDP
79.2	147.8	76.7	181.5

D. Multiple action recognition

In this section, we test the effectiveness of our models to identify multiple actions within a long video sequence. Specifically, we conduct experiments on continuous action

TABLE VII
COMPARISON WITH THE STATE OF THE ART METHODS ON S-KTH AND HOHA FOR ACTION SEGMENTATION

Methods	s-KTH	HOHA
DFW [25]	-	**45.72%**
HMM-MIO [24]	**71.2%**	-
Hoai et al. [8]	-	42.24%
SMM [23]	-	34.2%
ML-HDP(STIP)	70.7%	41.3%
w-ML-HDP(STIP)	67.4%	31.9%
ML-HDP(IDT)	83.4%	50.8%
w-ML-HDP(IDT)	79.2%	42.1%
ML-HDP(sTDD)	81.1%	53.4%
w-ML-HDP(sTDD)	75.4%	44.7%
ML-HDP(IDT+sTDD)	**85.3%**	**57.1%**
w-ML-HDP(IDT+sTDD)	82%	45.9%

segmentation and action localization. Then, a comparison with the state-of-the-art methods is presented for each task.

1) Continuous action segmentation: For jointly segmenting and recognizing continuous actions in a video, we classify each frame by using the method described in Section III-D. Two types of train sets are considered: one is strongly labeled and the other is weakly labeled. For the strongly labeled set, each pre-segmented video collected from train sets of KTH or HOHA contains exactly one action. For the weakly labeled set, we use train sequences of s-KTH and concatenated HOHA, where each video is annotated by a set of action labels with unknown order and unknown action boundaries. Our model learned with the weakly labeled set is denoted as w-ML-HDP.

The experimental results are compared with HMM-MIO [24] on s-KTH, and with DFW [25], SMM [23], and SVM-based model [8] on HOHA. These methods use strongly labeled data for training. From Table VII, ML-HDP significantly outperforms the others. Interestingly, w-ML-HDP with trajectory-based features yield the comparative performance, even though it deals with more outliers and noises due to the nature of weak supervision. Hence, the efficiency of w-ML-HDP is higher. Note that ML-HDP with STIP is comparable to the others also using STIP, such as DFW and HMM-MIO. On HOHA, DFW performs better than ML-HDP with STIP by 4.4%, but their algorithm requires the ordering constraint during training, while it is not necessary for our work. When evaluating different features under our framework, we can also draw a similar conclusion as action classification. In particular, trajectory-based features are significantly better than STIP. Compared to IDT, sTDD performs slightly worse on s-KTH, but it obtains the better result on HOHA. Finally, using combined features significantly boosts the accuracy.

2) Action localization: In the second scenario of multi-action recognition, spatiotemporal action locations are identified in a video. To validate the effectiveness of our method, we perform experiments on MSR-II and UCF101 datasets. For MSR-II, we follow the cross-dataset paradigm in [14].

In this experiment, besides target actions of each dataset, we also include negative action samples (e.g., irrelevant objects moving in background). The negative samples of MSR-II correspond to the "walking" videos of KTH. Meanwhile, the negative samples of UCF101 are the parts of training videos that do not contain bounding boxes. Then, by jointly learning multiple actions, the negative instances and the appearance of each action help us to better distinguish spatial locations of other target actions. The temporal segment is identified as the method described in Section III-D using optimal values of localization parameters (i.e., Q and Th). An action occurred in each frame of that segment is then spatially localized with bounding boxes. As shown in Figure 6, the visual words assigned to negative action labels are removed. Motivated by [22], visual words, those are spatially far from the region of interest, are pruned because they are uninformative and may negatively affect the localization accuracy. To do this, the interest points of a video segment are considered as nodes of the space-time graph. Then distances between interest points are measured to capture their relationship via established edges. Subsequently, we simply use the spatial distance for STIP feature, while we apply the trajectory distance as defined in [22] for IDT and sTDD. After that, we remove components of the graph containing a few nodes, i.e., 1 node for STIP and less than 5 nodes for IDT and sTDD. Finally, we draw bounding boxes based on eigenvalues and eigenvectors of the remaining points in each frame. For MSR-II dataset, since persons do not change their position, we first project locations of interest points onto the $xy-$ plane and then draw the bounding box based on eigenvalues and eigenvectors of projected points. As a result, we obtain a sequence of bounding boxes relevant to the subvolume (MSR-II) or tubes (UCF101) of target action. Figures 7 and 8 show some examples of localization results using our method on MSR-II and UCF101, respectively.

Fig. 6. Examples of pruning visual words. Left: a video frame contains negative label assignments (yellow) and "boxing" assignments (blue). Middle: the result after removing visual words assigned to negative labels. Right: the final result of "boxing" assignments after pruning visual words (at the right-bottom of the middle image) those are far from the region of interest.

Table VIII presents the performance comparison of action localization between ML-HDP with the state-of-the-art. As the usual practice, we set the overlap thresholds of 0.125 and 0.2 for MSR-II and UCF101, respectively. We can see that trajectory-based features performs much better than STIP by a large margin, and combining features further improves the localization performance. For MSR-II, ML-HDP outperforms benchmarked methods with the best result of 74.7%. On large-scale UCF101, the performance of ML-HDP is significantly higher than the trajectory-based proposal method (APT) [29], those contain a lot of false negatives due to a large number of generated action proposals. From Table VIII, the performance of ML-HDP is below recent deep-learning methods (e.g., [51], [52]), those generate strong frame-level proposals by the use of highly sophisticated two-stream CNN model. In summary, experimental results further demonstrate the effectiveness and the generality of ML-HDP for various recognition tasks.

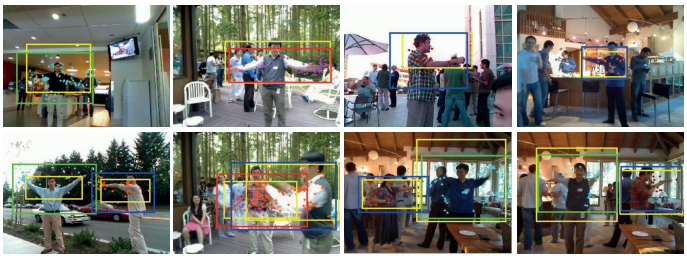


Fig. 7. Examples of localization results on MSR-II dataset. Yellow bounding boxes indicate ground-truth. Red, green, and blue bounding boxes indicate localization results of "hand-clapping"(magenta), "hand-waving"(cyan), and "boxing"(orange and maroon), respectively. Note: colors inside parentheses denote the associated topic assignment of that action.

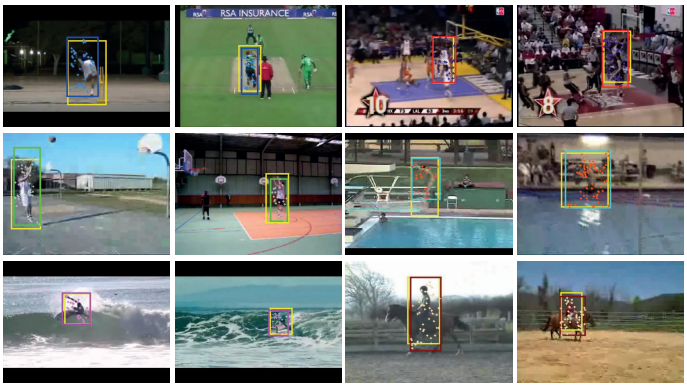


Fig. 8. Examples of localization results on UCF101 dataset. Yellow bounding boxes indicate ground-truth. Green, red, blue, cygan, maroon, and magenta bounding boxes indicate localization results of "basketball"(silver), "basketballdunk"(indigo), "cricketbowling"(dodger blue), "diving"(red orange), "horsediving"(wheat), and "surfing"(violet) respectively. Note: colors inside parentheses denote the associated topic assignment of that action.

V. Conclusions

In this paper, we have presented a framework based on the bag-of-words representation to recognize human actions in videos. We first proposed a ML-HDP model to jointly learn multiple high-level actions and lower-level motion units at different hierarchical levels. In particular, our ML-HDP model is based on the three-level representation: low-level visual words, atomic actions, and high-level actions. In our model, atomic actions are automatically discovered from the video data and act as mid-level representation, which enables us to improve the robustness of recognition model. We then interpreted the use of discovered atomic actions by conducting the experiments of topic visualization, which showed that our model works discriminatively and captures well the co-occurring motion patterns. Furthermore, we incorporated the notion of multiple action labels into our topic model to address the multi-labeling problem. Then, based on the inferred topic and label assignments, we proposed straightforward methods for three recognition tasks including: action classification, continuous action segmentation, and action localization. The extensive experimental results on standard datasets demonstrated the effectiveness and the generalization ability of our approach for various tasks, where we achieve the competitive performance compared to the state-of-the-art approaches.

As a future work, our proposed approach can be extended in several possible directions. First, with trajectory-based features extracted from the UCF101 dataset, the current learning process is still slow (around 15 days) due to a large number of Gibbs sampling iterations. Hence, we plan to develop the distributed algorithms [59] to further speedup the learning process and improve the scalability of ML-HDP. Alternatively, variational inference [10] is also worth to be examined as the faster solution for topic learning. Second, the spatiotemporal relationship between local features can be explicitly incorporated into our model to boost the performance of recognition tasks. Finally, since the applicability of ML-HDP is not limited to action recognition, we will investigate an extension of our model for solving other problems in computer vision such as object recognition or scene categorization. Thus, further study is needed to address these issues and provide a highly efficient recognition system.

TABLE VIII
Comparison with the state of the art methods on MSR-II and UCF101 for action localization

Methods	MSR-II	UCF101
Peng et al. [51]	-	**72.8%**
Saha et al. [52]	-	66.7%
APT [29]	**73.2%**	34.5%
Yu et al. [28]	61.3%	26.5%
Tubelet [27]	54.4%	-
SDPM [26]	35.8%	-
Cao et al. [14]	19.1%	-
ML-HDP(STIP)	34.3%	16.2%
ML-HDP(IDT)	70.2%	46.6%
ML-HDP(sTDD)	68.4%	50.2%
ML-HDP(IDT+sTDD)	**74.7%**	**53.1%**

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2016-0-00406, SIAT CCTV Cloud Platform).

References

[1] I. Laptev, "On space-time interest points," Int'l J. Computer Vision, vol. 64, no. 2-3, pp. 107-123, 2005.
[2] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 1-8, 2008.
[3] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "Action recognition with improved trajectories." Int'l J. Computer Vision, vol. 119, no. 3, pp. 219-238, 2016.
[4] H. Wang, A. Kläser, C. Schmid, and C. L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," Int'l J. Computer Vision, vol. 103, no. 1, pp. 60-79, 2013
[5] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," In BMVC 2009.
[6] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic annotation of human actions in video." Proc. IEEE Int'l Conf. Computer Vision, pp. 1491-1498, 2009.
[7] J.C. Niebles, C.W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," Proc. European Conf. Computer Vision, pp. 392-405, 2010.
[8] M. Hoai, Z.Z. Lan, and F. De la Torre, "Joint segmentation and classification of human actions in video." Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 3265-3272, 2011.

[9] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection." Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 2442-2449, 2009.

[10] D.M. Blei, A.Y. Ng, and M.I. Jordan "Latent Dirichlet allocation" J. Machine Learning Research, vol. 3, pp. 9931022, 2003.

[11] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. "Hierarchical dirichlet processes." J. Am. Statistical Assoc., 2006.

[12] Y.W. Teh and M.I. Jordan, Hierarchical Bayesian Nonparametric Models with Applications, Bayesian Nonparametrics: Principles and Practice, Cambridge Univ. Press, 2010.

[13] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," Proc. 17th IEEE Int'l Conf. Pattern Recognition, vol. 3, pp. 32-36, 2004.

[14] L. Cao, Z. Liu, and T.S. Huang, "Cross-dataset action detection," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 1998-2005, 2010.

[15] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," Proc. Ninth IEEE Int'l Conf. Computer Vision, pp. 1470-1477, 2003.

[16] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 2169-2178, 2006.

[17] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," Proc. IEEE Int'l Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 65-72, 2005.

[18] 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, vol. 34, no. 9, pp. 1704-1716, 2012.

[19] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," Int'l J. Computer Vision, vol. 105, no. 3, pp. 222-245, 2013.

[20] X. Peng, L. Wang, Y. Qiao, and Q. Peng, "Boosting vlad with supervised dictionary learning and high-order statistics," Proc. European Conf. Computer Vision, pp. 660-674, 2014.

[21] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," IEEE Trans. Image Processing, vol. 23, no. 2, pp. 810-822, 2014.

[22] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 1242-1249, 2012.

[23] Q. Shi, L. Cheng, L. Wang, and A. Smola, "Human action segmentation and recognition using discriminative semi-markov models," Int'l J. Computer Vision, vol. 93, no. 1, pp. 22-32, 2011.

[24] E.Z. Borzeshi, O.P. Concha, R.Y. Da Xu, and M. Piccardi, "Joint action segmentation and classification by an extended hidden Markov model," IEEE Signal Processing Letters, vol. 20, no. 12, pp. 1207-1210, 2013.

[25] K. Kulkarni, G. Evangelidis, J. Cech, and R. Horaud, "Continuous action recognition based on sequence alignment," Int'l J. Computer Vision, vol. 112, no. 1, pp. 90-114, 2015.

[26] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 2642-2649, 2013.

[27] M. Jain, J. Van Gemert, H. Jegou, P. Bouthemy, and C.G. Snoek, "Action localization with tubelets from motion," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 740-747, 2014.

[28] G. Yu and J. Yuan, "Fast action proposals for human action detection and search." Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 1302-1311, 2015.

[29] J. Van Gemert, M. Jain, E. Gati, and C.G. Snoek, "APT: Action localization proposals from dense trajectories," In BMVC, vol. 2, pp. 4, 2015.

[30] T. Hofmann, "Probabilistic latent semantic analysis," Proc. 15th Conf. Uncertainty in Artificial Intelligence, pp. 289-296, 1999.

[31] L.J. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," Proc. IEEE Int'l Conf. Computer Vision, pp. 1-8, 2007.

[32] N. Rasiwasia and N. Vasconcelos, "Latent dirichlet allocation models for image classification," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 35, no. 11, pp. 2665-2679, 2013.

[33] N.A. Tu, D.L. Dinh, M.K. Rasel, and Y.K. Lee, "Topic modeling and improvement of image representation for large-scale image retrieval," Information Sciences, vol. 366, pp. 99-120, 2016.

[34] X. Wang, X. Ma, and W.E.L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, no. 3, pp. 539-555, 2009.

[35] H.T. Thien, O. Banos, B.V. Le, D.M. Bui, Y. Yoon, and S. Lee, "Traffic behavior recognition using the pachinko allocation model," Sensors, vol. 15, no. 7, pp. 16040-16059, 2015.

[36] J.C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," Int'l J. Computer Vision, vol. 79, no. 3, pp. 299-318, 2008.

[37] Y. Wang and G. Mori, "Human action recognition by semilatent topic models," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, no. 10, pp. 1762-1774, 2009.

[38] M. Bregonzio, J. Li, S. Gong, and T. Xiang, "Discriminative Topics Modelling for Action Feature Selection and Recognition," In BMVC, pp. 1-11, 2010.

[39] J. Wang, P. Liu, M. FH She, A. Kouzani, and S. Nahavandi, "Supervised learning probabilistic latent semantic analysis for human motion analysis," Neurocomputing, vol. 100, pp. 134-143, 2013.

[40] C. Andrieu, N. De Freitas, A. Doucet, and M.I. Jordan, "An introduction to MCMC for machine learning," Machine learning, vol. 50, no. 1-2, pp. 5-43, 2003.

[41] A. Gilbert, J. Illingworth, and R. Bowden, "Action recognition using mined hierarchical compound features," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 883-897, 2011.

[42] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, " Beyond gaussian pyramid: Multi-skip feature stacking for action recognition.," Proc. IEEE Int'l Conf. Computer Vision, pp. 204-212, 2015.

[43] D. Tran and L. Torresani, "EXMOVES: classifier-based features for scalable action recognition," Int'l J. Computer Vision, vol. 119, no. 3, pp.239-253, 2016.

[44] R. Hou, A.R. Zamir, R. Sukthankar, and M. Shah, "Damndiscriminative and mutually nearest: Exploiting pairwise category proximity for video action recognition," Proc. European Conf. Computer Vision, pp. 721-736, 2014.

[45] M. Sapienza, F. Cuzzolin, and P.H. Torr, "Learning discriminative spacetime action parts from weakly labelled videos," Int'l J. Computer Vision, vol. 110, no. 1, pp.30-47, 2014.

[46] S. Sadanand and J.J. Corso, "Action bank: A high-level representation of activity in video," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 1234-1241, 2012.

[47] Y. Li, W. Li, V. Mahadevan, and N. Vasconcelos, "Vlad3: Encoding dynamics of deep features for action recognition," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 1951-1960, 2016.

[48] B. Fernando, P. Anderson, M. Hutter, and S. Gould, "Discriminative hierarchical rank pooling for activity recognition," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 1924-1932, 2016.

[49] M. Jain, J.C. van Gemert, and C.G. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 46-55, 2015.

[50] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 4305-4314, 2015.

[51] X. Peng and C. Schmid, "Multi-region two-stream R-CNN for action detection," Proc. European Conf. Computer Vision, pp. 744-759, 2016.

[52] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzlion, "Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos," in BMVC, 2016.

[53] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 596-603, 2014.

[54] M.S. Ryoo and J.K. Aggarwal, "UT-interaction dataset, ICPR contest on semantic description of human activities (SDHA)," In IEEE Int'l Conf. Pattern Recognition Workshops, vol. 2, pp. 4, 2010.

[55] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, pp. 2929-2936, 2009.

[56] K. Soomro, A.R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild", CRCV-TR-12-01, 2012

[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition" arXiv preprint arXiv:1409.1556, 2014.

[58] J. Zhu, A. Ahmed, and E.P. Xing, E.P., "MedLDA: maximum margin supervised topic models," J. Machine Learning Research, vol. 13, pp.2237-2278, 2012.

[59] D. Newman, A. Asuncion, P. Smyth, M. and Welling, "Distributed algorithms for topic models," J. Machine Learning Research, vol. 10, pp.1801-1828, 2009.

[60] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," Proc. European Conf. Computer Vision, pp. 581-595, 2014.
[61] M. Sekma, M. Mejdoub, and C.B. Amar, "Human action recognition based on multi-layer fisher vector encoding method," Pattern Recognition Letters, vol. 65, pp. 37-43, 2015.

Nguyen Anh Tu received the B.S. degree of Electrical and Electronics Engineering from Ho Chi Minh City University of Technology, Vietnam, in 2010, and the Ph.D. degree of Computer Science and Engineering from Kyung Hee University, Republic of Korea, in 2018.

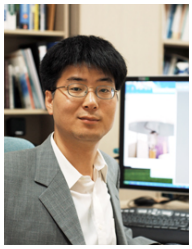
He is currently a Post-Doctoral Research Fellow with Data and Knowledge Engineering in the Department of Computer Science and Engineering at Kyung Hee University, Republic of Korea. His current research interests include computer vision, machine learning, image retrieval, and big data processing.

Thien Huynh-The received his B.S. degree of Electronics and Telecommunication Engineering and M.Sc. degree of Electronic Engineering from Ho Chi Minh City University of Technology and Education, Vietnam in 2011 and 2013, respectively. In 2018, he received the Ph.D. degree of Computer Science and Engineering from Kyung Hee University, Republic of Korea. He is awarded with Superior Thesis Prize by KHU.

Currently, he is a Post-Doctoral Research Fellow with Ubiquitous Computing Laboratory at Kyung Hee University, Republic of Korea. His current research interests are digital image authentication, computer vision, and machine learning.

Kifayat Ullah Khan received his B.S., M.S., and Ph.D. from Gomal University, Pakistan, University of Greenwich, UK, and Kyung Hee University, Republic of Korea in 2005, 2007, and 2016 respectively. He worked as a Post-Doctoral Fellow in the Department of Computer Science and Engineering at Kyung Hee University, Republic of Korea from September, 2016 to February, 2018.

Currently, he is working as a Senior Lecturer in Institute of Computing and Information Technology, Gomal University, Pakistan. His research interests include database, data warehousing, graph mining, and graph summarization.



Young-Koo Lee received his B.S., M.S., and Ph.D. in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea in 1988, 1994, and 2002, respectively. From 2002 to 2004, he was a Post Doctoral Fellow Advanced Information Technology Research Center (AITrc), KAIST, Republic of Korea, and a Postdoctoral Research Associate at the Department of Computer Science, University of Illinois at Urbana-Champaign, USA.

Since 2004, he has been a Professor at the Department of Computer Engineering, College of Electronics and Information, Kyung Hee University, Republic of Korea. His research interests are ubiquitous data management, data mining, activity recognition, bioinformatics, on-line analytical processing, data warehousing, database systems, spatial databases, and access methods.