UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

VIDEO SCENE UNDERSTANDING:

SEMANTIC-BASED REPRESENTATION, TEMPORAL VARIATION MODELING AND MULTI-TASK LEARNING

Negar Rostamzadeh

Advisor

Prof. Nicu Sebe

Università degli Studi di Trento

April 2017

Abstract

One of the major research topics in computer vision is automatic video scene understanding where the ultimate goal is to build artificial intelligence systems comparable with humans in understanding video contents. Automatic video scene understanding covers many applications including (i) semantic functional complex scene categorization, (ii) human body-pose estimation in videos, (iii) human fine-grained daily living action recognition, (vi) video retrieval, and genre recognition. In this thesis, we introduce computer vision and pattern analysis techniques that outperform the state of art of the above mentioned applications on some publicly available datasets. Our major research contributions towards automatic video scene understanding are (i) introducing an efficient approach to combine low and high-level information content of videos, (ii) modeling temporal variation of frame-based descriptors in videos, and (iii) proposing a multitask learning framework to leverage the huge amount of unlabeled videos. The first category covers a method for enriching visual words that contain local motion information but they lack information about the cause of the motion. Our proposed approach embeds the source of a generated motion in video descriptors and hence induces some semantic information in the employed visual words in the pattern analysis task. Our approach is validated on traffic scene analysis as well as human body pose estimation applications. When employing an already-trained off-the-shelves model over an unseen dataset, the accuracy of the model usually drops significantly. We present an approach that considers low-level cues such as the optical flow in

the foreground of a video to make an already-trained, off-the-shelves, pictorial deformable model work well on a body pose estimation working well for an unseen dataset. The second category covers methods that induce temporal variation information to video descriptors. Many video descriptors are based on global video representations, where, frame-based descriptors are combined to a unified video descriptor without preserving much of the temporal information content. To include the temporal information content in video descriptors, we introduce a descriptor, namely, the Hard and Soft Cluster Encoding. The descriptor includes how similar frames are distributed over a video timespan. We present that our approach yields significant improvements on the human fine-grained daily living action recognition task. The third category includes a novel Multi-Task Clustering (MTC) approach to leverage the information of unlabeled videos. Our proposed method is on human fine-grained daily living action recognition application. People tend to perform similar activities in the similar environments. Therefore, a proper clustering approach could determine patterns of fine-grained activities during some learning process. Our proposed MTC approach rather than clustering the data of each individual separately, capture more generic patterns across users over the training data and hence leads to remarkable recognition rates. Finally, we discuss opportunities for future applications of our research and conclude with a summary of our contributions to video understanding.

Keywords

Computer Vision, Machine Learning, Video Understanding, Enriching Video Representations, Modeling Temporal Variation

Acknowledgements

My Ph.D. was an amazing and unforgettable journey. Not only I entered to the world of research, but also I started my Ph.D. in a new country and a totally new environment. All these years I had the support and love of my best friend, my husband, Mojtaba. We started the path of Ph.D. together and he was always kept me motivated. No matter what we were going through, he was always giving me his love and support. So many things in our life have changed during the path of our Ph.D. but he was always there for me. My next big thanks go to my supervisor, Nicu. He helped me entering the world of research. His kind smile was always a great motivation for me to keep going. He was always supportive of my decisions and giving me advice when I needed it. If it was not because of him and amazing flexibility in our group, Mhug, under his direction, I was not able to explore the world of research and industry as much as I did. He was not only a great supervisor but also a great friend to me. Huge thanks go to Jasper Ujilings, who was my colleague and was co-supervising me on a few projects in Mhug. I learned a lot from his experiences. During my Ph.D. I spent a summer at the Multimedia and Vision lab at the University of Queen Marry in London under the supervision of Yiannis Patras. He was always full of ideas and helped me a lot with ideas and exploring different directions. My next journey was at MILA (former LISA) in the University of Montreal under the supervision of Aaron Courville. I was lucky that I had his support. He was a great sample of a person who was deeply exploring the problems, ideas, and solution and at the same time enjoying the research path. I had a great

time doing research under his direction. MILA is directed by Yoshua Bengio. I learned from him to think big and he also made me really interested in open-sourcing and sharing knowledge for the sake of science. Seeing Yoshua not only caring about the world of research but also about the humanity and posting his ideas about important events in the world, made me think more about what is happening in the world and how we can have an impact for good. In MILA, I was co-supervised by Chris Pal. He was a really nice person and always could think of interesting applications to apply interesting research ideas on them. My next journey was at Google in the Machine Intelligence team directed by Blaise Aguera y Arcas. Working on his team was one of my best experiences during my Ph.D. He was an amazing leader and an awesome person. He nurtured a diversified and happy team of researchers, engineers, and designers. If one day, I want to lead a research or industrial team, I will try to do it the way that he is doing. His caring about all team members and endless support meant a lot to me. I also want to thank my manager at Google, Seymour, my co-hosts, Iwona and Emily, who were so supportive of me, Andrea, Francesca and Anna at the UNITN doctoral school, Myriam, Linda and Angela at MILA. My next thanks go to all my friends and colleagues in Mhug (University of Trento), MILA (University of Montreal), MMV(Queen Mary University of London) and Google. I was so lucky to be surrounded with many amazing people who helped me through my Ph.D. path. Then my friends who were always supportive of me. Farnaz, Faranak and Shahrzad who were unconditionally supporting me and were always there for me. My friend and kung fu master, Ostad Mahnoush, Shirin. My amazing researcher friends who make me being a proud woman in Tech by their love and support, Jamie, Amy, Timnit, and Adriana. Finally my amazing family, my mom, my dad and my brother, Pouya for their love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Automatic video scene understanding and activity analysis are active research topics in computer vision. The interest in video analysis is motivated by the promise of important applications in several fields, such as patient monitoring, ambient assisted living, security (*e.g.* video recording in banks) and traffic congestion analysis (*e.g.*, at junctions) and more recently self-driving cars. The automatic video analysis systems are proposed against the traditional approaches that employed humans to review the videos and extract the interesting activities and events from them. The extreme goal of such investigations is to design a system that is comparable with human in understanding, summarizing and indexing a video scene. In this framework, semantic information is the information that a human can directly preserves by observing a scene. Such as the information that shows which objects are presented in the scene. In terms of this definition, semantic information is on the opposite side of the low-level cues such as local-motion or local-appearance. Particularly in this work we address three interesting video scene understanding scenarios:

1. Semantic functional complex scene categorization.

2. Human body-pose estimation in videos.

3. Human fine-grained daily living action recognition.

4. Video retrieval, genre recognition and human action recognition in wild and realistic scenarios.

To address the **first scenario**, we propose approaches to understand and divide a complicated scene into meaningful regions based on the functionality of the objects present in the scene (*QMUL Junction*[1] and *MIT traffic* dataset[2]).
To address the **second scenario**, we propose approaches to embed temporal information into a dynamic frame-based pictorial model, for the task of pose estimation in videos.
For addressing the **third category**, we proposed approaches to distinguish between fine-grained activities that differ only slightly in motion and appearance and are performed indoor in scene with static background (*ADL Rochester* dataset). For the *fourth category*, we propose approaches on multimodal datasets (MediaEval, Blip10000), and classifying the genre of movies from a combination of audio and video data and classifying complex activities that are performed outdoors in scenes with dynamic background (such as *UCF50* and *UCF101* datasets).

Each of these scenarios is interesting and difficult for the following reasons:

(i) Understanding complicated traffic scenarios containing pedestrians and cars moving in many different directions with a variety of velocities is a very complex task. In this work, we employ a novel approach for semantic functional categorization of such scenes and show results on 2 standard crowded traffic scenes: QMUL Junction and MIT traffic datasets.

(ii) Human body pose estimation in realistic scenarios with a noisy background is a very challenging task due to occlusions and range of body movements (*Pose in the Wild* dataset). In video scenarios, with static background and high resolution such as *ADL Rochester* dataset, estimating the

[1] http://www.eecs.qmul.ac.uk/~sgg/QMUL_Junction_Datasets/Junction/Junction.html

[2] www.ee.cuhk.edu.hk/xgwang/MITtraffic.html

pose is easier but it is challenging due to another reason: ''For old age homes, nursing homes soon personal care robots will be deployed in near future. Human detection, pose estimation with high accuracy is a requirement. Also the models used should not be memory intensive."[3]

(iii) Analyzing daily living scenarios is a challenging task. First of all, in such a scenario, different activities differ only slightly in motion and appearance. In some cases, the differences in appearance of the subjects who perform the same task are more evident than the difference in activities. Moreover, one activity can be performed in many different ways, while two different activities may be performed in a very similar manner with respect to motion and appearance. For example, dialing and answering the phone are activities only slightly different in terms of hand movements. Particularly, if we consider the Activity of daily living Rochester dataset, the difference between two activities in most cases is limited to taking *phone, banana or knife* from the *table, shelf, or refrigerator* and doing slightly different other activities (*e.g.* eat snack and drink water).

(iv) Analyzing sports activities performed in a complex and dynamic background is also extremely challenging. We showcase our approaches on the *UCF50* and *UCF101* dataset which consists of videos that are taken from youtube. This dataset is very challenging due to large variations in camera motion, object appearance, pose and scale, viewpoint, cluttered background and illumination conditions. The video clips of the different activity may share some common features, such as the same person, similar background, similar viewpoint, and so on. We are also applying our approaches on Multimodal data (*Blip10000* and *MediaEval* datasets), which are challenging due to the variation in the data, the length of videos and the realistic nature of videos.

[3]wikipedia on applications of human body pose estimation approaches

1.1 Problem Statement

Here we address three main problems in the state-of-the-art and briefly explain how we address them:

(i) Most of the works in the literature use motion information but ***discard the high-level information coming from the source of motion***. We address this gap particularly in (1) human activity recognition scenarios and (2) semantic functional scene categorization.
In human action recognition scenario, the parts of the body involved in a task should be considered, otherwise, some activities may not be distinguishable (*e.g.* push and kick). In Traffic scenarios, without knowing the source of a detected motion, it is difficult/impossible to semantically categorize the scene. For an anomaly detector, it is also important to know if a motion belongs to a pedestrian or a car. It is being recognized that using the high-level information coming from the detectors allows a deeper comprehension of the scene. The reason why the use of detectors in video scene analysis has not yet been widespread is because in many video analysis cases lots of different detectors should be applied or the occlusions prevent obtaining a good detection performance (*e.g.*, the case of *UCF101*, *MediaEval* datasets). Moreover in the case of human pose estimation, many scenarios, such as pedestrian monitoring, suffer from occlusions and the performance of the detectors is not always good enough to be relied upon. However, in a daily living scenario, the person is monitored by a camera in a controlled environment and the body is clearly visible and mostly not occluded. In the case of Traffic analysis, sparse information that comes from detectors provide a deeper comprehension of the scene. Traffic analysis sparse information that comes from detectors provide a deeper comprehension of the scene. We aim to exploit high-level information and exploit them in more scenarios. For doing this an efficient and accurate body

pose estimator is required. In the case of body pose estimation, a significant drop in accuracy has been observed when a detector is trained on one dataset and it is evaluated on a different one [81]. The reason is that for some cases there are not enough samples in the training set. As the detector gives more priority to the positive samples of the training set, the chance of detecting uncommon (*w.r.t* positive samples) body poses decreases. A possible solution to this is to set the body pose ground-truth for the new dataset and re-train the classifier. However, this procedure is very expensive and requires a consistent delay every time a new dataset has to be analyzed. Instead of training another classifier on the new dataset, we propose to use the already trained classifier, but we provide some additional information from the new dataset. In the case of body pose estimation in videos, we propose to use long-term motion information(*Dense Trajectories* and *Optical flow*) and embedding ***sufficient amount*** of this information into our pose-estimator framework. Sufficiency of motion cues in our approach is evaluated by our proposed Motion Relevancy descriptor, which we will thoroughly explain in the second chapter of the thesis.

In the case of complicated scenarios where the number of semantic classes is not defined, it is impossible to know a priori the number of detectors to be used to capture the nature of objects. In addition applying too many different detectors does not seem to help the action recognition process. For example in the UCF101 sport dataset, even for the clips that show the same activity, there are different kinds of objects that are present in the scene. In such cases, we apply a combination of low-level cues to capture extra information.

(ii) In video analysis, another important problem is ***how to adequately capture temporal information***. Until recently, most video retrieval systems relied mostly on single representative video frames where time is ignored

for efficiency reasons [108]. Recent work simply accumulates features over a whole video sequence [2, 22, 99]. Such accumulation may capture more information but also mixes up the information, disregarding the appearance variation over time. For example, when a car approaches and then turns a corner, there are first straight movements followed by turning movements. It is clear that both types of movements do not happen at the same time. Because of this, we want to have a representation which keeps this distinction and model the distribution of features in time.

In this third chapter of this thesis, we propose two novel video representations to capture ***temporal variation***. Specifically, we firstly propose to use the ***fisher representation on global features*** [69], which was recently introduced for modeling the spatial variation in a Bag-of-Words framework [17].

Then we propose the use of Temporal Fisher Kernel and Temporal VLAD representations on several bag-of-words representations of different low-level features to model the temporal variation. Finally, we propose ***a new Temporal Soft Cluster encoding*** to capture the features distributions in time.

(iii) Classifying Activities of Daily Living (ADL) is a challenging and important task because some of the videos that are associated with various classes differ only slightly. In addition, very similar videos may present two different activity classes. On the other hand, we address the task of ADL classification, when some of the labels of the activities are not available. We propose a ***Multi-Task Clustering*** approach to classifying partially labeled data.

1.2 Contribution of this work

The main contributions of this theis are detailed as follows.

1.2.1 Efficient Combination of Low and High level Features in Videos

Enhanced semantic descriptors for functional scene categorization

In this work[4], we present a novel approach which combines semantic information with low level features extracted from a complex video scene. The proposed method for video scene understanding relies on a bag-of-words approach, in which, typically, visual words contain information of local motion, but information regarding what generated such motion is discarded. Instead, in our framework, the semantic information is embedded in the visual words and it allows to automatically obtain semantic categorization of the scene. We show the effectiveness of our method in a traffic analysis scenario: in this case, two main semantic classes, pedestrians and vehicles, are discovered. We present our results on two publicly available datasets (1) MIT Traffic dataset (2) Queen Mary Junction dataset.

Human Pose Estimation considering Dense Trajectories Length

In this work[5], we present an approach for human body pose estimation in videos. Our approach is built upon a *Deformable Part-based Model (DPM)*. We introduce a new descriptor namely, *Dense Trajectories Length (DTL)*. Embedding *DTL* in a *DPM* framework, biases the wrists locations to areas with significant motion. The contribution of *DTL* is optimized in an iterative way and by the help of *Motion Relevancy (MoR)* descriptor. In fact, we investigate the effectiveness of motion information in predicting body poses in videos. For example embedding the motion information in some frames help the pose estimation and in some frames, it may downgrade the results. We address this issue and optimize the contribution of motion to reach an optimized pose estimation in all video frames. We evaluate our approach on two publicly available datasets (1)

[4]This paper is published in the International Conference on Pattern Recognition (ICPR) 2012

[5]This work is part of my research output during my internship at MMV lab, University of Queen Mary of London. This research is mainly supervised by Prof. Ioannis Patras

Pose in the Wild and (2) ADL Rochester Pose datasets.

1.2.2 Modeling Temporal Variation of Features in Videos

Time Matters! Capturing Variation in Time in Video using Fisher Kernels

In video global features are often used for reasons of computational efficiency, where each global feature captures information of a single video frame. But frames in video change over time, so an important question is: how can we meaningfully aggregate frame-based features in order to preserve the variation in time? In this work[6] we propose to use the Fisher Kernel to capture variation in time in video. While in this approach the temporal order is lost, it captures both subtle variation in time such as the ones caused by a moving bicycle and drastic variations in time such as the changing of shots in a documentary.

Our work should not be confused with a Bag of Local Visual Features approach, where one captures the visual variation of local features in both time and space indiscriminately. Instead, each feature measures a complete frame hence we capture variation in time only.

We show that our framework is highly general, reporting improvements using frame-based visual features, body-part features, and audio features on three diverse datasets: We obtain state-of-the-art results on the *UCF50 human action recognition dataset* and improve the state-of-the-art on the *MediaEval 2012 video-genre benchmark* and on the *Activities of Daily Living Rochester* dataset.

Cluster Encoding For Modeling Temporal Variation in Video

Classical Bag-of-Words methods represent videos by modeling the variation of local visual descriptors throughout the video. In this approach they mix variation in time and space indiscriminately while these dimensions are fundamentally different. Therefore, in this work[7] we present a novel method for video

[6] This paper is published in the ACM Multimedia 2013

[7] This paper is published in International Conference of Image Processing (ICIP) 2015

representation which explicitly captures temporal variation over time. We do this by first creating frame-based features using standard Bag-of-Words techniques. To model the variation in time over these frame-based features, we introduce Hard and Soft Cluster Encoding, novel techniques to model variation inspired by the Fisher Kernel [1] and VLAD [2]. Results on the *Activities of Daily Living Rochester* [3] and *Blip10000* [4] datasets show that our method yields improvements of respectively 6.6% and 7.4% over our baselines. On Blip10k we outperform the state-of-the-art by 3.6% when using only visual features.

1.2.3 Recognizing Activities of Daily Living in Videos

Daily Living Activities Recognition via Efficient High and Low Level Cues Combination and Fisher Kernel Representation

In this work[8] we propose an efficient method for activity recognition in a daily living scenario. At feature level, we propose a method to extract and combine low- and high-level information and we show that the performance of body pose estimation (and consequently of activity recognition) can be significantly improved. Particularly, we propose an approach extending the pictorial deformable models for the body pose estimation from the state-of-the-art. We show that including low level cues (*e.g.* optical flow and foreground) together with an *off-the-shelf* body part detector allows reaching better performance without the need to re-train the detectors. Finally, we apply the Fisher Kernel representation that takes the temporal variation into account and we show that we outperform state-of-the-art methods on *Activities of Daily Living Rochester* dataset.

[8]This paper is published in International Conference on Image Analysis and Applications (ICIAP) 2013

It's all about Habits: Exploiting Multi-Task Clustering for Activities of Daily Living Analysis

Motivated by applications in areas such as patient monitoring, telerehabilitation and ambient assisted living, analyzing activities of daily living is an active research topic in computer vision and image processing. In this research[9], we address the problem of everyday activity recognition from unlabeled data proposing a novel Multi-Task Clustering (MTC) approach. Our intuition is that, when analyzing activities of daily living, we can take advantage of the fact that people tend to perform the same actions in the same environment (*e.g.* people working in an office environment use to read and write documents). Thus, even if labels are not available, information about typical activities can be exploited in the learning process. Arguing that the tasks of recognizing activities of specific individuals are related, we resort on multitask learning and rather than clustering the data of each individual separately, we also look for clustering results which are coherent among related tasks. Extensive experimental results show that our method outperforms several state-of-the-art approaches by up to 11% on the Rochester activities of daily living dataset.

1.3 Outline

The remaining part of this thesis is organized as follows. In Chapters 2, we present our work on combining low- an high-level information in videos for the tasks of *Semantic Functional Categorization of traffic scense* and *Human body-pose estimation in wild and daily living activities*. In Chapter 3, we present our work on *modeling the temporal varition in videos*. In Chapter 4, we study *Daily Living Activities Analysis in Videos*. Finally, conclusions are drawn in Chapter 5.

[9]This paper is published in International Conference on Image Processing (ICIP) 2014

Chapter 2

Efficient combination of low and high level features in videos

Visual surveillance in dynamic scenes is an important area, strongly driven by many potential applications such as traffic scene understanding and behaviour monitoring. One important research question in video understanding is how to enrich the descriptors with high-level semantic information or low-level information and in an efficient framework. This chapter[1], consists of the following studies:

- In section 2.1, we present a novel approach which combines semantic information obtained from two classes of detectors (*pedestrian and vehicle detectors*), with low level features extracted from a complex video scene (*motion*). In this research, we rely on a bag-of-words framework.
  In our framework, the semantic information is embedded in the visual words and it allows to automatically obtain semantic categorization of the scene.
  We present the effectiveness of our framework on two publicly available datasets: (1) MIT traffic and (2) QMUL Junction datasets.

[1]The research presented in section 2.1 is published in the International Conference of Pattern Recognition (ICPR2012) [146] and the presented research in section 2.2 is a part of my research outputs during an internship at MMV lab in the University of Queen Marry of London, where I was supervised by Dr. Ioannis Patras

- In section 2.2, we present a novel approach for the task of human body-pose estimation in videos. In a Deformable Part-based Model (*DPM*), we enrich the appearance descriptors (*hog*) in the semantic class of *wrists*, by information derived from the Length of Dense Trajectories (*DTL*) passing through different location of the scene. The contribution of *DTL*, then is optimized in an itterative framework. We present our results on two publicly available datasets: (1) ADL Rochester pose estimation and (2) Pose in the Wild datasets.

2.1 Enhanced semantic descriptors for functional scene categorization

Recently, non-object-centric approaches for the analysis of dynamic scenes have gained popularity and have been proven effective for many application scenarios, *e.g.* extraction of salient activities [145], scene semantic segmentation [48], *etc*. In particular, in the case of complex scenarios with many targets and occlusions, they are preferred to the classic object detection/tracking schema because they are not subject to the problem of broken trajectories or to the curse of dimensionality when trying to consider the spatio-temporal correlation between many targets. In a nutshell, non-object-centric methods rely on low level cues (*e.g.* local motion and foreground) that are analyzed in a bag-of-words framework. Firstly, a visual codebook of the scene is generated, where visual words generally encode information about the local motion in the scene. Secondly, the video is divided into (i) short clips (or spatial patches) and for each of them a bag-of-visual-words is build. Starting from this bag-of-words representation of a video stream, statistical models such as Probabilistic Topic Models (PTM) are deployed in order to mine (i), the typical patterns of behavior and the anomalies from the scene or (ii), the spatial segmentation of the scene, where patches with similar motion behavior are assigned to the same semantic area. This approach has shown to be robust to noise; however, there are some limitations. Switching from an object-centric to a non-object-centric perspective, the method gains in robustness w.r.t noise and to broken trajectories; however, at the same time, the information about the semantic of the objects causing the detected motion (*e.g.* cars, pedestrians, *etc.*) is discarded. As a consequence, it is difficult to reason about topics extracted as they do not always correspond to high-level description of the scene according to a human observer. In this work, we focus on this aspect and propose the use of enhanced semantic descriptors, as a step towards filling the semantic gap between object-centric and non-object-centric

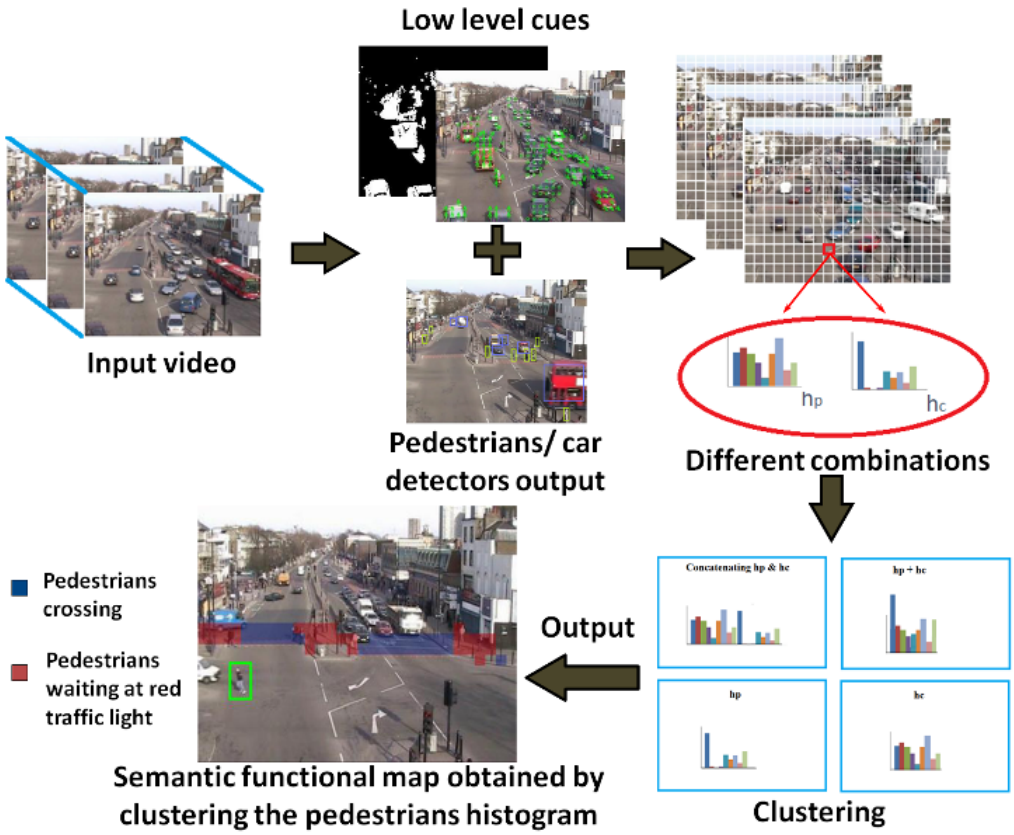


Figure 2.1: An overview of the pipeline we propose for analyzing traffic scenarios.

approaches. We show our results on publicly available datasets and compare them with recent related work.

2.1.1 Related Work

In the last decades, the bag-of-words paradigm has been widely adopted firstly in still images analysis [17, 25] and then in dynamic scenes analysis [126, 65, 59]. This paradigm is composed of two main steps: (i) codebook generation (ii) bag-of-words formation and clustering. In detail, the first phase corresponds to the definition of the scene descriptors and thus it strongly influences the final results and the effectiveness of the method. In other words, if the features we extract from our scene do not properly represent the activities occurring in

the scene, or some important information is discarded in this phase, the topics extracted during the second phase may not be a representative synopsis of the scene observed. In particular, by discarding semantic information in this preliminary step of the analysis, the final topics extracted may not correspond to the high level segmentation generated by a human observed. This problem has been addressed by [49] with Object Bank for still images analysis: the intuition is that scene descriptors are combined with information of the objects in the scene. While obtaining state-of-the-art performance, this method is found to be very expensive when a large number of categories is considered (as in the case of still images). In the case of video scene analysis, much effort has been devoted to enrich the descriptors with additional information beyond motion. Messing *et al.* [59] used descriptors which encode information of both local motion and appearance, in order to distinguish between actions with similar motion but different appearance (*e.g.* eating snack from eating banana) and between actions with similar appearance but different motion (*e.g.* peeling banana from eating banana). Additionally, they used sparse information provided by a face detector to augment their features with relative position information. Shitrit *et al.* [101] track people by linking sparse information of people detection into tracklets. This approach is proven to be more effective in term of robusteness and complexity then recursively track from frame to frame. Sangmin *et al.* [66] detected functional objects in traffic scenes (*e.g.* delivery track) by using features which encode relation and actions w.r.t the scene context. Chen *et al.* [15] proposed the use of motion features (MoSIFT) to encode static image appearance features together with motion information.

Based on [124], Papageorgiu *et. al* [67] proposed a general object detection scheme using Haar wavelets and SVMs and applied it for face, pedestrian and car detection. Other works focused specifically on pedestrian [125] or car [149] detection: Viola *et al.* [125] were able to detect pedestrians at very small scales (up to 20×15 pixels) by using both appearance and motion information; Zhu

et al. [149] devised a car detection scheme exploiting global structure and local texture features. In order to reduce human efforts in the labeling task, methods based on semi-supervision or active learning have been proposed [83, 64]. Finally, a relevant work [133] proposed a method for adapting a generic pedestrian detector to specific traffic scenes, exploiting multiple cues such as size and motion.

2.1.2 Our method

Similarly to Turek *et al.* [118], our aim is to perform a semantic segmentation of the scene, where the scene element categories are primarily defined by their behavior, rather then their appearance or shape. In particular, Turek *et al.* [118] use a hierarchy of motion features. They divide the scene into several patches and for each patch a codebook histogram is formed. Patches are then clustered according to their histograms' similarity and the patches belonging to the same cluster are associated to the same functional category. Differently from them, in our work the semantic information of the entity causing the motion is embedded in the descriptor. This is obtained by combining information provided by the pedestrian and vehicle detectors with local motion information.

For the pedestrians and cars detector we rely on [125]. For low level cues extraction, we use Lucas Kanade algorithm [114] for optical flow combined with dynamic Gaussian-Mixture background model [109]. Motion vectors are quantized into 8 possible directions while patches with only static points and sufficient foreground are considered associated to a static event (*e.g* pedestrian stopped at red traffic light waiting to cross the street). Then, for each patch we build 2 histograms of 9 bins each (8 bins identify motion and 1 bin is for static events).

In details, our method is formulated as follows. We divide our scene into $N_x \times N_y$ patches. Then, for each of these patches $p_{i,j}$, where $i = 1, ..., N_y$ and $j = 1, ..., N_x$, we build 2 histograms of cars h_C and pedestrians events h_P

by analyzing a video sequence of at least 30 minutes of length. In particular, we consider a couple of frames i and $i - I_s$ at a time, where I_s is the step for optical flow computation. For each frame i, we compute the corresponding foreground mask F^i and, still on frame i, we run the car and the pedestrian detectors. We define $B_P^i = \{b_1, ..., b_{N_P^i}\}$ and $B_C^i = \{b_1, ..., b_{N_C^i}\}$ as the resulting bounding boxes found, localizing respectively pedestrians (P) and cars (C), The bounding box $b_k = (x, y, w, h)$, with $k = 1, ..., N_K^i$ and $K = \{P, C\}$, is defined by the coordinate of its upper left corner (x, y) and its size (w, h). N_K^i is the number of items found, which varies at each frame i. Every time a static or a motion event is detected, it is counted as an occurrence in the corresponding h_P or h_C histogram, depending on if it is included, respectively, in one of the bounding box from B_P or B_C. Once the histograms of the patches are built, we cluster them in order to obtain the spatial segmentation of the scene. Similarly to [118], we label cells by using k-means and mean-shift. We obtain 4 spatial segmentations by considering for clustering:

- 18 bins *semantic* histograms, obtained by concatenating h_P and h_C
- 9 bins *non semantic* histograms, obtained by summing h_P with h_C,
- 9 bins *only-car semantic* histograms h_C
- 9 bins *only-pedestrian semantic* histograms h_P

The effect of differently combining h_P and h_C are discussed in the experimental session. Once the semantic maps of the scene are built, they can be used for further analyses of events in the scenes.

It is worth noting that in application with crowded scene scenarios the object centric paradigm based on detection/tracking is not reliable because of the high risk of tracking failures and the curse of dimensionality. However, the use of sparse information provided by the object detectors combined with low level features allows to exploit the advantages of both non object centric and object

Table 2.1: Details on experimental setup

	n^o frames	fps	video duration	frame size $W \times H$	patch size $s_W \times s_H$	n^o patches $N_x \times N_y$
MIT Traffic	162000	30	$1h$ 30'	720×480	15×15	48×32
QMUL Junction	90000	25	$1h$	360×288	12×12	30×24

centric methods; respectively these advantages include (i) robustness to noise and reduced computational complexity on one side and, on the other one, (ii) encoding semantic information of the entities interacting in the scene.
Another advantage of the method is that we can perform an analysis of the scene at two different levels of optical flow. This allows to detect objects at different speeds, like in the case of cars and pedestrians. In fact, by using a low value of I_s, the shift measured of a slow moving object (*e.g.*, a pedestrian) is close to zero and thus it is detected as a static event. On the other side, by increasing the value of I_s, we can detect the motion of slow moving object, but the motion cues extracted for high speed object tend to be very noisy. In our experiment we set $I_s^C = 5$ and $I_s^P = 10$.

2.1.3 Results

We show our results on two public datasets: MIT Traffic dataset[2] and QMUL Junction dataset[3]. Sample frames extracted from these two datasets are shown in Fig. 2.2; details on the datasets and the experimental setup are summarized in Table 2.1. Our semantic segmentation method's results are visually displayed in Fig. 2.3 and 2.4, respectively for the MIT Traffic and QMUL Junction datasets.

[2] www.ee.cuhk.edu.hk/~xgwang/MITtraffic.html
[3] www.eecs.qmul.ac.uk/~jianli/Junction.html

Figure 2.2: Dataset used. (Left) MIT Traffic and (Right) QMUL Junction dataset

MIT Traffic dataset

Visual inspection of our segmentation results on the MIT Traffic dataset (Fig. 2.3) against the path model reported and exploited in [133], shows that our proposed method performs comparably while being extremely simple and cheap to implement. It requires, in fact, the availability of a detector for the objects of interest in the scene and makes use of simple and well-known features, such as optical flow: for both the requirements, extensive literature and code are available. Figure 2.3(b) shows that with our method, by using only non semantic descriptors, some areas like zebra crossing are not distinguished. By using semantic descriptors the areas associated to pedestrians or cars are better distinguished, as it can be seen in Fig.2.3(a). Still, as observable from Fig.2.3(c) and Fig.2.3(d), clustering the two histograms separated may provide the best solutions. In (c) we distinguish between different cars lanes, associated to the different main motion directions, while in (d) we can distinguish between two different conceptual areas for pedestrians: zebra crossing (highlighted in blue) and sidewalks (in violet). In Fig.2.3(d) additionally we show an example of anomaly that can be detected by using the semantic pedestrian map generated, *i.e.* a pedestrian walking outside the pedestrian area (in a red frame).

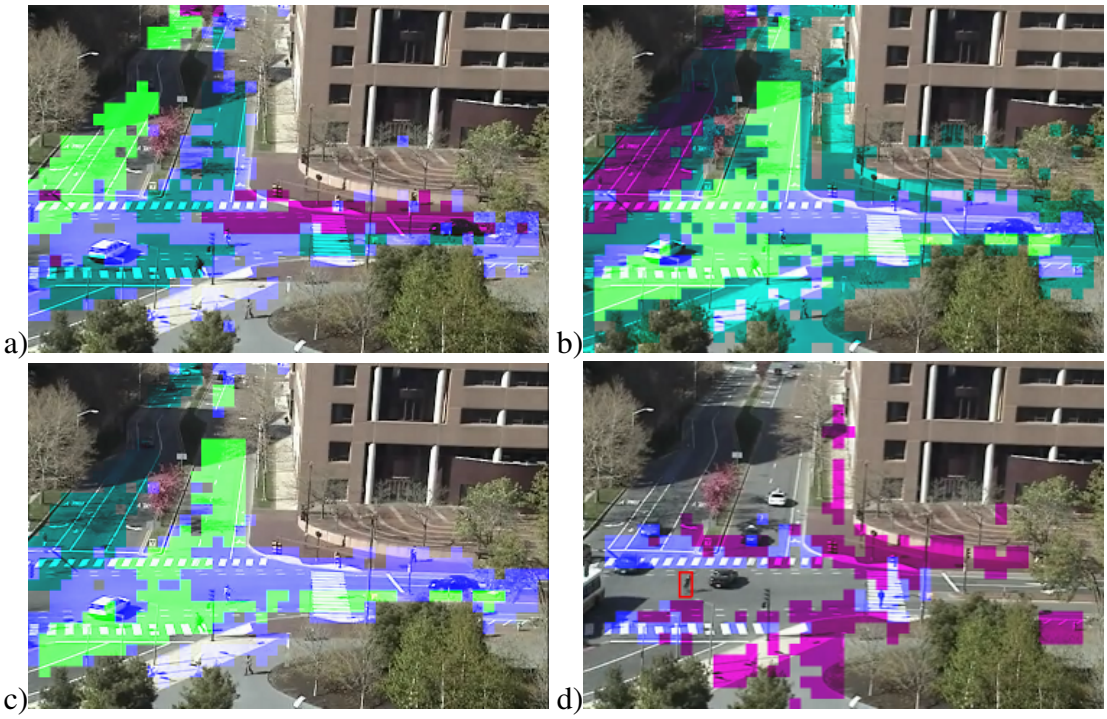


Figure 2.3: MIT dataset. Scene segmentation based on (a) semantic, (b) non-semantic, (c) car semantic and (d) pedestrian semantic descriptors.

QMUL Junction dataset

In Fig. 2.4 a sample case of a pedestrian detected in an unusual position w.r.t the pedestrian semantic map of the scene is depicted. Due to its highly cluttered background, we could not get a reliable pedestrian detector to work with this dataset. Still, relying on ground-truth knowledge on some pedestrian positions we are able to infer, thanks to semantic scene segmentation, anomalies in the pedestrian behavior (e.g., crossing outside of zebras). Interestingly, observing the semantic pedestrian map shown in Fig 2.4(a), we notice that the two discovered clusters correspond to two conceptually different areas: i) zebra crossing, shown in blue, and ii) the waiting areas at the intersection between sidewalks and zebra crossings when the pedestrian traffic light is on red (shown in red). Our intuition is that the algorithm proposed by [124] could benefit in terms of speed, to some extent, and, more significantly, of a reduction in false positives by discarding detection sub-windows containing less than a given amount of foreground (shown in Fig.2.4(b)). Additionally, the semantic maps obtained by using the ground truth on pedestrians could be used for the same goal of [133] (i.e., filtering out false positives from automatically gathered training data in a novel scene). In future works we plan to present an extension of such algorithm in this direction.

2.1.4 Conclusions

We proposed a novel method which relies on the classical bag-of-words model but instead of ignoring the semantic information as it is typically done in the literature we include it in our descriptors. We presented results on two traffic datasets where two semantic object categories (cars and pedestrians) are present. While proving their effectiveness on anomaly detection tasks (*e.g.* pedestrian on the car lane), semantic maps can be useful for other tasks as shown in literature (*e.g.* automatic selection of training set [133]). Further work will

Figure 2.4: (Left) Anomaly. Pedestrian detected outside associated semantic map. (Right) corresponding Foreground mask.

involve considering other scenarios and extending the current object detector to exploit semantic maps for improving performance.

2.2 Human Pose Estimation considering Dense Trajectories Length

The problem of human body pose estimation in both images and videos has drawn increasing attention during the last few years [14, 91, 142]. The interest in human pose estimation is mainly motivated by its applications in human action recognition [82, 84, 143], sign language translation [14] and video understanding [38]. Recent studies show that improvement in Human Pose Estimation contributes to better Action Recognition performance [29, 84, 143]. In fact, involving the information derived from high-level detectors, decreases the ambiguity in fine-grained action recognition.

Human Pose Estimation is challenging due to body-range of motion, degrees of freedom and shape from different views, invisibility or barely visibility of some of the body-parts, clutter backgrounds and variety of clothing.

According to the state-of-the-art approaches, the head and the neck are usually very well localized by applying the classical pose estimation models in single images and videos [16, 142]. However, the wrists, elbows and shoulders are much more challenging. Movements and poses of these challenging body-parts are more important for recognizing the human actions.

Some recent approaches devoted their effort on improving the localization of challenging body-parts, (*i.e. wrist, elbow*) [14, 16]. Similar to [14, 16], we concentrate on improving the upper/lower arm (*i.e*; shoulders, elbows and wrists) accuracy rates.

We assume that *When a person is performing an action, the wrists move more than the other parts of the body*.

Rostamzadeh *et. al.* [84] states involving the number of flows between every two sequential frames can improve the accuracy of *wrists* localization more than the other body-parts. Based on this assumption, we introduce a descriptor that model the number of dense trajectories that pass through every locations

of a certain frame. We name the descriptor as *Dense Trajectory Length map* descriptor or shortly *DTL*. Then using *DTL* with spatial-based features of each frame, helps enhancing the wrist localization. Finally using dynamic programming, the localization of elbows and shoulders are refined upon achieving a better wrists location.

The initial assumption of associating wrists locations to the most dynamic places of a scene, in practice, has some limitations that we address in this study: (1) During any activity, there may be some fraction of a second that the wrists are not moving much, or their length of $3D$ movement is not reflected well in $2D$ because of the camera view. (2) There may be clutter motion coming from the background, that could potentially mislead the detection (*e.g: sea waves in the background*). The camera movement may also affect the motion pattern for some frames. (3) There are also some frames, which the pose estimation relying on the initial image-based features yields almost a perfect accuracy rate. In these cases involving the information coming from the *DTL* not only doesn't improve the localization rate but also may worsen it.

To avoid downgrading the high quality predictions obtained from our initial image-based model, we estimate the relative importance of the *DTL* over the *spatial* features.In order to estimate the importance of the *DTL* feature, we introduce a feature inspired from the *SIFT* descriptor but over the temporal domain. Our new feature is obtained from (i) an initial wrist localization together with (ii) the motion cues (*i.e.* optical flow) and is entitled *motion relevancy (MoR)*. MoR estimates *how much the motions around predicted wrists are relevant to a valid prediction*. As a summary our two main contributions in this paper are as following:

- We use the *DPM* framework on the frames of a video but with a different cost function at the leaf nodes that are associated with the wrists. In our proposed cost function for wrists, we add a term, as a modified *Dense Trajectories Length map (DTL)*, which leads the wrists more towards areas

with significant motion. Finally, the whole pose estimation framework optimizes all body joints locations simultaneously using dynamic programming.

- We propose an iterative approach to optimize the contribution of *DTL* in our proposed *DPM* framework. Consequently, we improve the localization accuracy of wrists, elbows and shoulders.

We apply our approach on two publically available datasets, (1) Rochester Activities of Daily Living (ADL) pose [84] and (2) Pose in the Wild [16]. The rest of the paper is organised as follows: In section 2.2.1, we discus about the related works on pose estimation in images and videos. Section 2.2.2 presents the details of our approach. In section 2.2.3, we present the effectiveness of our approach through the experiments. In section 2.2.4, we draw a conclusion.

2.2.1 Related work

Yang and Ramanan [140] proposed an image-based *Deformable body Part Model*. They consider body-parts as deformable parts that are relied upon a Tree-based model and using dynamic programming optimizing the location of different body joints. Our *3D* human pose estimation model is built upon the *2D-DPM* [140] pose estimation model.

In general, most of the *state-of-the-art* Human Pose Estimation approaches in videos can be divided into *two major categories*:

1. The first category includes methods that consider there is at least one frame with correctly detected body-parts. Then the detection of the joints in the other frames becomes a tracking problem [76].

2. The second category covers methods that optimize body-joints localization jointly in every short sequence of frames [150].

In the *first category* either (i) the body pose is manually annotated in a few frames [11, 13] or (ii) an approach is developed to automatically recognize the frames that the initial image-based pose estimator works reasonably good on them [76, 150]. These approaches (namely, **strike a pose**) basically assume that in every performed action, there are at least a few frames that estimated poses, are easily recognizable regarding a few predefined poses [76, 150]. For the next step, these approaches worked on the tracking problem.
In the *second category*, either (i) an image-based approach is applied as an initial estimation and then the estimated poses are adjusted according to the temporal information [84] or (ii) the estimated body poses in consequent frames are jointly optimized in both spatial and temporal domains [14, 16], which is usually time consuming and computationally expensive.

Our approach falls into the first branch of the second category. In our approach, we assume that *when a person is performing an action, the wrists move more than the other parts of the body*. The assumption is basically motivated by the future work suggestions of [16], and [84] that states involving the number of flows between every two sequential frames can improve the accuracy of *wrists* localization more than the other body-parts. Based on this assumption, we introduce a descriptor that models the number of dense trajectories that pass through every location of a certain frame. We name the descriptor as *Dense Trajectory Length map* descriptor or shortly *DTL*.

Many state-of-the-art approaches on Pose-Estimation learn a structured model by employing a set of low/mid-level features. Some recent cascade approaches [91, 92], learn a coarse to a fine sequence of discriminative models using the information being present in space and time. In the cascade approaches a top-down model is trained [91, 92]. This top-down model (1) narrows down the searching space for a perfect pose candidate, and (2) avoids extracting expensive features unless the perfect candidate pose is not detected before the final

stage. Thus, the cascade approaches are successful models particularly in terms of efficiency.

Apart from the efficiency and computational cost of feature extraction, some features are better in capturing some types of poses, when the same features provide some redundant information for the other frames/images. For example, skin map helps wrist localization when people wear long sleeve. However, this may not be a very successful feature when people wear a swimming suit or short sleeve shirts. The given example suggests the contribution of some features for the pose estimation task over various frames shall be considered.

In the cascade approaches [91, 92], including or excluding a feature, is dependent on a binary classifier. In our approach, instead of predicting the presence of a descriptor (namely, *DTL*) using a binary classifier, we measure the importance of the given descriptor in a given frame.

In some of the state-of-the-art pose-estimation approaches on videos, an off-the-shelves image-based pose-estimator(*e.g. DPM*) is refined by (1) employing the information content of the motion direction (tracking) [104] and (2) improving the consistency of detected poses on consequent frames [102].

In this study, we use the *Deformable Part-based Model (DPM)* [140] framework but we employ a different cost function at the leaf nodes that are associated with the wrists in order to bias the wrist location towards areas of significant motion. Then using the dynamic programming, all body joints are simultaneously localized and their positions are optimized.

2.2.2 Proposed Approach

In this section, we present our method for *Pose Estimation in videos*. Body-pose estimation in videos plays a key role in important computer vision applications such as Human Action Recognition. A good estimation over the wrists location, ease understanding the performed action [84, 143]. Therefore, it would be worthy to put the focus on improving wrist localization. In addition, wrists

are the most challenging parts to detect in pose estimation problem [14, 16, 93]. *Rostamzadeh et al* shows [84], leading the detection of body-parts to the regions with high motion, improves the wrists localization significantly more than the other body-parts. *Cherian et. al.* [16] propose, as an extension to their work, to enrich the wrist model by encouraging the alignment of wrists position to the regions with high motion.
Involving the number of flows in a small region around every candidate locations (as a descriptor), help improving the pose estimation from an image-based approach [84].
According to [16, 84], wrists move more than the other parts. Inspired by [16, 84], we employ the *DTL* feature to refine the initial localization of wrists.
First, instead of considering only the flow between 2 consecutive frames, we consider the length of the dense trajectories that path through different regions of every given frame by introducing the *DTL* feature map. We use the Deformable Part-based Model (*DPM*), due to its computational efficiency and reasonable performance [142]. The deformable mixture-of-parts model, or shortly ***DPM model***, is proposed for single images [142].

We use *DPM* framework, but we add an additional scoring term (*i.e. DTL score*) at the wrists. The *DTL* term, biases the estimation towards areas of significant motion in a period of time. Then using the dynamic programming all body-joints are simultaneously optimized.
DTL helps localization of wrists and consequently all body-joints. The *DTL* is defined as *number of flows that pass through a certain location over a short period of time*. The details of computing the *DTL* descriptor is presented in the following subsections.
Then relative importance of involving *DTL* descriptor in a *DPM* framework is estimated automatically through a set of classifiers. More specifically, in every step of an iterative process, a binary classifier checks whether the estimated-pose can be improved by increasing the importance of wrist-movements or

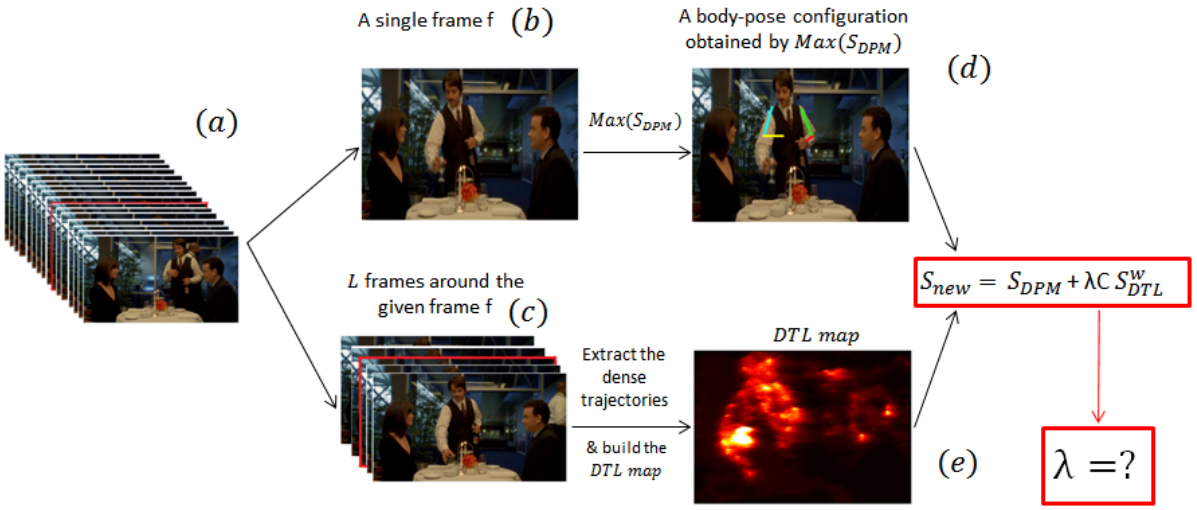


Figure 2.5: The general framework: (a) presents the input video, and the frame (b) is a single frame from the video a. (d) presents the pose estimation result that is obtained from the (a) by the image-based *DPM* pose estimation method. (c) is a video with the length of L frames around the frame (b). (e) is the *DTL* map that is obtained from the dense trajectories over the video (c). (e) presents the regions of the scene that has more dynamics with hot color. In the formula, S^{w}_{DTL} presents the value of the *DTL* map that is only involve on top of the wrist local scores of the *DPM* approach. λ is not a fix number and it is obtained in an iterative way.

not. The iteration then is continued till the classifier returns a negative value. Thereof, we control the impact of the *DTL* feature on the modified *DPM* model. In order to train the classifiers, we calculate the *Motion Relevancy* or shortly ***MoR*** descriptor around the localized wrists in every iteration-steps. This descriptor models the distribution of movements around the dominant flow direction. We employ the *MoR* descriptor to evaluate whether the motion around the detected wrists on that step reveals meaningful information to help the pose estimation or not.

Deformable Part-based Model

We build our pose estimation approach on the image-based *deformable mixture of part model (DPM)* [142] that has a *pictorial structure*. In a pictorial structure human body is modelled as a graph, *G=(V, E)*. Nodes *(V)* present body-parts (*e.g; elbows, head*) and edges (springs between parts) model the geometric consistency between body-parts *(E)* [26].

In a pictorial structure body is presented as an ideal template. A *deformation* is defined as replacement of springs, while the structure of the model is preserved. *Deformations* over the ideal structure present possible body-configurations.

According to [26], each possible body configuration is given a score. The ideal estimated pose in the image I (with the size (w, h)) is computed through optimizing the body-configuration score as follows:

$$S(P, I) = \sum_{i \in V} \phi(p_i, I) + \sum_{(i,j) \in E} \psi(p_i - p_j) \tag{2.1}$$

$\phi(p_i, I)$, represents local score of the part i in the location p_i of the given image I.

Where $p_i = (x_i, y_i)$ and $\forall i \in \{1, 2, ..., n\} : x_i \in \{1, 2, ..., w\}, y_i \in \{1, 2, ..., h\}$, $n =$ the number of body-parts).

The local score, $\phi(p_i, I)$, presents the appearance similarity of any possible locations p_i and part i of the body. Formula 2.1, according to [142], local appearance model is a HoG descriptor extracted from a block around the pixel location p_i and the given joint at the center of the part i.

Pairwise score (*deformation model*), $\psi(p_i - p_j)$ is the relative location of the joints i and j. The pairwise score, $\psi(p_i - p_j)$, measure the probability of geometric configuration of the *(i,j)* pairs [26].

P is the set of corresponding body-part locations: $P = \{p_1, p_2, ..., p_n\}$.

In the *DPM* model [142], each body-part is associated to one of t possible types (mixture components). We follow the same setting and formulations of

[142], but for the simplicity we skip presenting details related to the types in the formula, as it is not related to our contributions.

$S(P, I)$ presents the body-configuration score. Maximizing $S(P, I)$ regarding to the placements of body-parts and mixture components, gives the optimal body-configuration.

Moreover, following [142] we search for human in different scales in an image pyramid. So the possible locations of body-parts, p_i, is in a grid with a specific resolution of the image I.

In the *DPM* model, the body relational graph is presented as a tree. There is no loop in a Tree representation and every joint except the root has only one parent. Thereby, we optimize $S(P, I)$ efficiently with dynamic programming in polynomial time [142].

We define T_{p_i} as a sub-tree of the general body tree that has the root p_i.

In this way, $\forall i \in \{1, ..., n\} : S(T_{p_i}, I)$, is the score for a tree with the node k as its root:

$$S(T_{p_i}, I) = \phi(p_i, I) + \sum_{j \in kids(p_i)} [S(T_{p_j}, I) + \psi(p_i - p_j)] \tag{2.2}$$

Since every part has only one parent, a message is passed from every child, j to its parent i. We define the children of the part i as $kids(p_i)$. Leaves of the tree, p_k, $k \in TreeLeaves$ don't have any kids and their corresponding sub-tree is presented by only a single node:

$$S(T_{p_k}, I) = \phi(p_k, I) \tag{2.3}$$

We iterate over all body-parts starting from the leaves and move upstream to the root part (*i.e ; head*, part 1 according to Figure 2.2.2).

As it is presented in figure 2.2.2.c, we give the body number 1 to the root. General body configuration is presented by T_{p_1} and optimizing the $S(T_{p_1}, I)$ using dynamic programming gives the optimum body configuration.

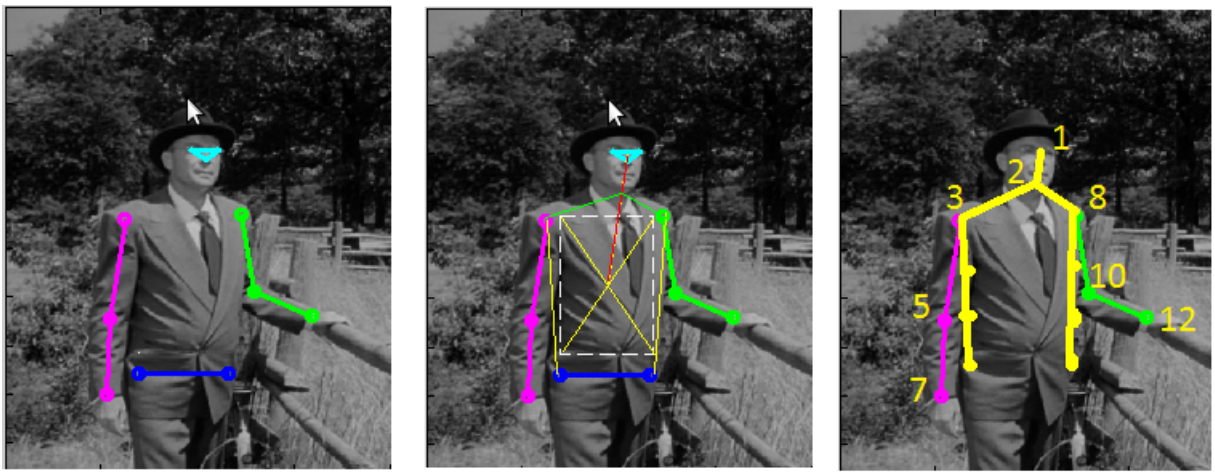


Figure 2.6: A body-configuration sample annotation from the FLIC dataset: (a) presents the pose annotation (b) drawn lines for augmenting the training data (c) presents the final augmented ground-truth over training data. The numbers of torso and the numbers between wrist-elbow and elbow-shoulder are not presented in the figure to increase the clarity of main parts.

The inference corresponds to maximizing the score function of the root, $S(T_{p_1}, I)$ over placements and types of different parts. So the optimized root score gives the optimum body-configuration.

Exploiting the Dense Trajectory Length (DTL) feature in a Deformable Part-based Model (DPM)

In this section, we explain the procedure of building the *DTL* feature map. In order to build the *DTL* feature map for every frame, we first extract the dense trajectories feature [130] of the given video.

In this section, we present a summary over the (i) dense trajectory extraction (ii) building the *DTL* map and (iii) linear interpolation of the *DTL* map. To extract the dense trajectories we follow the setting proposed by [130]: We densely sample feature points on a spatial grid by $w \times w$ pixels blocks. Then any spacial grid that is build has a block sizes as an integer multiplication of $w \times w$ blocks.

For different spacial scales dense optical flow field is computed on all spacial locations of different grid scales. Then we densely track all the sampled points through every part of the video for the fixed length of L frames without additional costs on top of the second stage.

After extracting the dense trajectories for all different scales, we build the *DTL* feature map. Given a frame-sequence $I = \{I_1, I_2, ..., I_k\}$, we make the *DTL* map for the frame f ($f \epsilon \{1, ..., k\}$) as follow: Consider $LT^f_{i,j}$, as the location (i, j) of the frame f. ($1 \leqslant i \leqslant h_f$, $1 \leqslant j \leqslant w_f$, while h_f and w_f are respectively the height and width of the frame f)

Then we extract the dense trajectories with the trajectory length of L frames for the given frame-sequence (video). $TR^s_{i,j,f}$ is the euclidean length of the trajectory that is started from the frame s and path through the location $LT^f_{i,j}$ in frame f, while $s \leqslant f \leqslant s + L$. Based on the initial setting this trajectory stops at the frame $s+L$. The *DTL feature* for frame f in the location (i, j) is then computed as follows:

$$D^f_{i,j} = \sum_{s=f-L+1}^{s=f+L-1} TR^s_{i,j,f} \tag{2.4}$$

In this formulation for $s > k$ and $s < 1$, $TR^s_{i,j,f} = 0$.

Equation 2.4 presents the values of *DTL* map for any given pixel location (i, j). As the dense trajectories are extracted on a grid space with the size of w, no value is available for the pixels that are not on the grid. Thus we first consider the values of these pixels as 0. Then, we apply a median filter with size $w \times w$ on the *DTL* map to get the final *DTL* map.

Figure 2.7: Sample frames from the pose-in-the-wild and videopose2 datasets are presented in the left and the *DTL* of these frames are presented in right.

The *DTL-based* pose estimation formulation

In this subsection, we detail our new formulation for pose estimation problem in videos. The new formulation employs the *DTL* map to *adjust the local score of the wrists*. Indeed, we involve the information related to the dynamic of the scene (through the *DTL* map) into the wrist model.
Then we optimize the impact of the *DTL* map for both wrists. Finally the elbows and shoulders optimized locations are also updated through dynamic programming with the optimized wrists location detections simultaneously.
Since the wrists are among the leaves of the pictorial model, for any given frame f, the wrist model in Equation 2.3 changes as follow :

$$S(T_{p_k}) = \phi(p_k, f) + \lambda_f C \cdot DTL^f_{p_k} \tag{2.5}$$

where, k is either left (l_w) or the right wrist (r_w). The $S(T_{p_k})$ score is computed for all possible locations in the corresponding grid of all different scales. λ_f controls the contribution of the *DTL* on pose estimation, and is automatically set using the method proposed in 2.2.2. C is a constant number that is empirically set to help us looking for the optimum λ_f in the $[0, 1]$ interval:

$$\lambda_f \in \{0, 0.05, 0.1, 0.15, 0.2, ..., 1\}$$

After the wrist model is refined, by employing the formula 2.5 instead of formula 2.3, the optimized locations of the rest body-parts are also updated by dynamic programming as follows:
By employing the formula 2.5 instead of formula 2.3 for the wrists, wrists scores are refined. The local scores of other body-parts, p, are also updated through collecting messages from the children of p including the wrists. Once messages are passed to the root part (part number 1), maximizing the $S(T_{p_1})$ (as it is stated in formula 2.2), gives the optimum body configuration over the body-parts locations and types.
λ_f is a variable that estimates the contribution of the *DTL* feature map for the

frame F.
Choosing the λ_f specify the relative importance of the dynamic and appearance of the scene in a given frame of the video.
Estimating the λ_f is dependent on a new descriptor, that we name it Motion Relevancy descriptor, or shortly ***MoR***. MoR_f helps optimizing the λ_f parameter by modelling the motion pattern between the frame f and its next frame.

Extracting the Motion Relevancy descriptor

In this subsection, we explain the procedure of extracting the *Motion Relevancy descriptors* (MoR) of the frame f with a given λ.

MoR^f_λ is an orientation invariant descriptor that model the motion pattern around the wrists. MoR^f_λ descriptor is inspired by $SIFT$ and $SIFT3D$, but over the temporal domain.

For every given λ, a body-configuration is obtained by optimizing the $S(T_{P_1})$ as explained in subsection 2.2.2. In the obtained body-configuration, we name the pixel locations of left and right wrists respectively w_l and w_r. Then the MoR descriptor of each wrist is extracted independently, as MoR_l and MoR_r. Finally the MoR^f_λ is computed as a combination of MoR_l and MoR_r. We first explain the procedure of extracting the MoR_r around the right wrist (w_r). MoR_l is also extracted with the same procedure but around the left wrist (w_l). Then we explain the process of building the MoR^f_λ when we have MoR_r and MoR_l.

Extracting the MoR_r :

1. We build a grid with 9 patches around the w_r as it is presented in Figure 2.2.2 (b). The pixel location of w_r is in the center of the central patch of the grid G_r. The size of every patch of the grid is $d \times d$ pixels. We name the central patch as C and the rest from top-left to center-left in spiral way as B_1 to B_8.

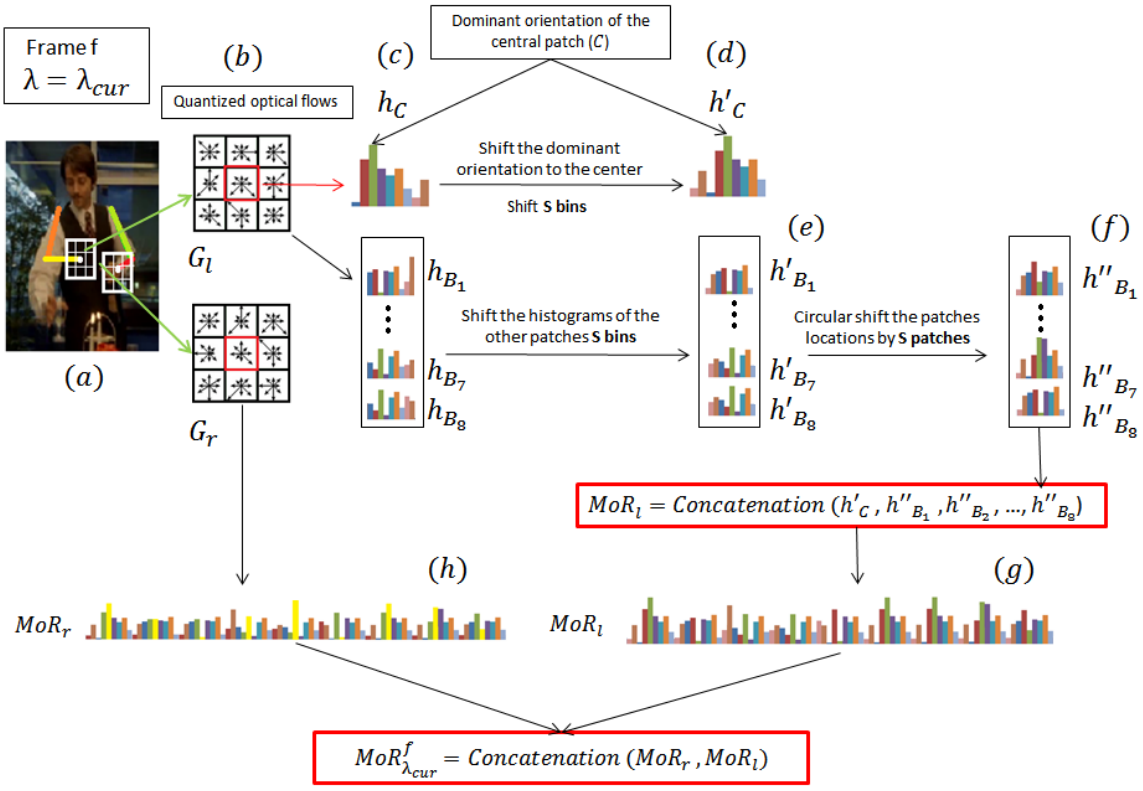


Figure 2.8: Motion relevancy descriptor: Given the frame F and $\lambda = \lambda_{cur}$, using the formula presented in Figure 2.2.2, estimated pose presented in (a) is obtained. Around the detected wrists a bounding box with 9 small patches with the fixed length of d is drawn. Optical flow between the given frame (a) and its adjacent frame is computed and quantized (b). (c) presents the histogram of quantized motion for the central patch of the left wrist (h_C) and the rest of the patches ($h_{B_1}, ..., h_{B_8}$). We circular shift the h_C bins to place the dominant bin in the middle (in the figure we shifted it by $+2$ bins). So we obtain h'_C (d). Then we circular shift the histograms of the rest patches (h_{B_i}) with the same shifting value (in this sample $+2$) to obtain h'_{B_i} presented in (e). Afterwards, we circular shift the location of every patch by $+2$ patches, and h''_{B_i} are obtained (f). Finally the MoR_l is built with the way that we present in (g). MoR_r is also computed with the same way but around the right wrist (h). The final descriptor then is computed by concatenating MoR_l and MoR_r.

2. We extract the optical flow between frame f and $f+1$ in the patches of the grid G_r. Note that if the frame f is the last frame of the video, we extract the flows between $f-1$ and f. The flows corresponding to different patches are quantized into 9 different directions (*i.e*: each of these directions has 40 degrees difference in orientation with the adjacent orientation). Then we build a histogram per patch that presents the number of flows in every directions. We name the histograms belonging to different patches as h_C and h_{B_i}. h_C and h_{B_i} ($i \in \{1, ..., 8\}$) presents the histogram respectively for the central patch and the patches around it.

3. We shift the bins of the histogram h_C, circularly to the right s **bins** in the way that the dominant bin goes to the center of the h_C (or to the bin number 5 as it is presented in Figure 4). We call the updated histogram of h_C as h'_C.

4. In this step, we right circle shift the bins of histograms h_{B_i} ($i \in \{1, ..., 9\}$) by s **bins**. We call the updated histograms as h'_{B_i}.

5. In this step, we relocate the patches h'_{B_1} to h'_{B_8} by shifting them s patches right circularly. More in details, shifting the h'_{B_i} by s patches, means placing the patch B_i into the location of B_j, where $j = mod(s+i, 8)$. Then we name the histograms belonging to the new placements as h''_{B_i} ($i \in \{1, ..., 8\}$).

6. Finally the MoR_r is obtained by concatenating the obtained histograms in the following order:

$$MoR_r = \left[h'_C, h''_{B_1}, ..., h''_{B_8}\right]$$

After acquiring both MoR_r and MoR_l independently with the explained procedure, we concatenate them to build the MoR^f_λ descriptor.

Meta-learning for optimal image-adaptive λ_f selection

As explained above, the λ_f value, that ranges in $[0, 1]$, controls the importance of including the DTL information. To automatically tune λ_f, we use a meta classification scheme in which a bank of classifiers are applied iteratively in order to decide whether the optimal value of λ_f has been reached or not. Therefore, we train 20 binary classifiers ($\mathcal{C}_k$) in correspondence to the k^{th} entry in $[0, 0.05, .., 0.95]$ as the candidate value of λ_f. This section details how we train each classifier and how we employ the predictions to set the λ_f value.

Train the $\mathcal{C}_k$ classifier

We assume a candidate value for λ_f as the k^{th} ($1 \leq k \leq 20$) entry in $[0, 0.05, .., 0.95]$, thereof $\lambda_k = (k-1) \times 0.05$ and we train $\mathcal{C}_k$ in correspondence to λ_k. To produce the positive and negative examples on our training data, our framework is as follows:

We first estimate body poses on the training data using the formulation 2.2 and 2.4 for all the training frames and for all values of λ_k. Therefore, for a given train frame (f_{tr}) we obtain an estimated pose ($P^{\lambda_k}_{f_{tr}}$). We calculate the MoR descriptor for the estimated poses for all the training frames.

Using the evaluation protocol explained in [28], we look for the smallest $\lambda_{f_{tr}} \in [0, 0.05, .., 0.95, 1]$ that provides the best average accuracy over wrists, elbows and shoulders on the frame f_{tr}. Let us denote the optimum $\lambda_{f_{tr}}$ as $\lambda^{opt}_{f_{tr}}$ and the corresponding optimum pose as $P^{opt}_{f_{tr}}$.

For the t^{th} training frame (f_{tr}), upon obtaining the set P_t, we associate a binary label to the corresponding MoR descriptor of a estimated pose (P_{f_{tr},λ_i}) depending on the λ value (=λ_i). The MoR descriptor of P_{f_{tr},λ_i}:

(a) is given a 1 label if $\lambda_k < \lambda_{f_{tr}}^{opt}$

(b) is given a 0 label if $\lambda_k = \lambda_{f_{tr}}^{opt}$

Given the training samples and their associated labels we train a Naïve Bayes classifier($\mathcal{C}_k$).

Prediction of the optimum λ

Given a sample test frame, f_{ts}, we obtain the pose estimation ($P_{f_{ts}}^{\lambda_k}$) in correspondence to each candidate λ_k value in $[0, 0.05, .., 0.95]$. We employ the DTL information to calculate the MoR descriptor for each estimated pose. We use $\mathcal{C}_k$ to classify the obtained descriptor with λ_k and hence we obtain a binary sequence with the length of 20 in correspondence to the twenty λ_f values. Then we apply a median filter with the length of 3 on the output of the 20 classifiers. The λ_f is set to the correspondent λ_k value for which the output of $\mathcal{C}_k$ is 0 and the previous outputs are equal to 1.

Algorithm 1 Updating λ

Require: $\lambda = \lambda_0$ with current value equals to 0. A sequence of options $\Lambda = \lambda_0, ..., \lambda_k, ..., \lambda_K$
Update flag $F =$ TRUE

Ensure: Updated proximity thresholds λ_{opt}

```
while λ! = λ_K and F do
    calculate MoR_{f_cur}
    apply the k-th classifier C_k on MoR_{f_cur}
    if then C_k returns TRUE
        k := k + 1
        λ = λ_k
    else
        F = FALSE
    end if
end while
```

2.2.3 Experiments and Results

In this section, we report the results and implementation details on two publically available datasets: (i) Pose in the Wild (ii) ADL Rochester datasets.

Pose in the wild dataset

Pose in the wild dataset is a challenging dataset, containing 30 sequences, with about 30 frames each, extracted from the Hollywood movies ''Forrest Gump", ''The Terminal", and ''Cast Away". Frames all annotated with upper-body poses. The dataset contains realistic poses in outdoor scenes, with background clutter, severe camera motion and bodypart occlusions [16]. The dataset is publicly available.[4]

Pre-training DPM model on the FLIC dataset Following [16], we train our model on the FLIC dataset [135]. We show that a small trick on data augmentation can improve the baseline by a good margine. Our trick is to just connect the middle of the torso to the middle of the head pose triangle and then consider the one third of then point the one third of the distance as neck. The visual details of the data augmentation on the FLIC dataset annotation is preesnted in Fig 2.2.2. In fact, we are just changing the data annotation to make it similar to the employed data annotation in [142] for upper body pose estimation.

Evaluation on the Pose in the Wild [16]

To make our results comparable with the state-of-the-art approaches on this dataset, we rely on [16, 91, 93], and evaluate our pose estimation approach by percentage of the predicted joints within a fixed distance of the groundtruth joints. We present the results in the range of 15 to 40 pixels distances.

[4] http://lear.inrialpes.fr/research/posesinthewild/

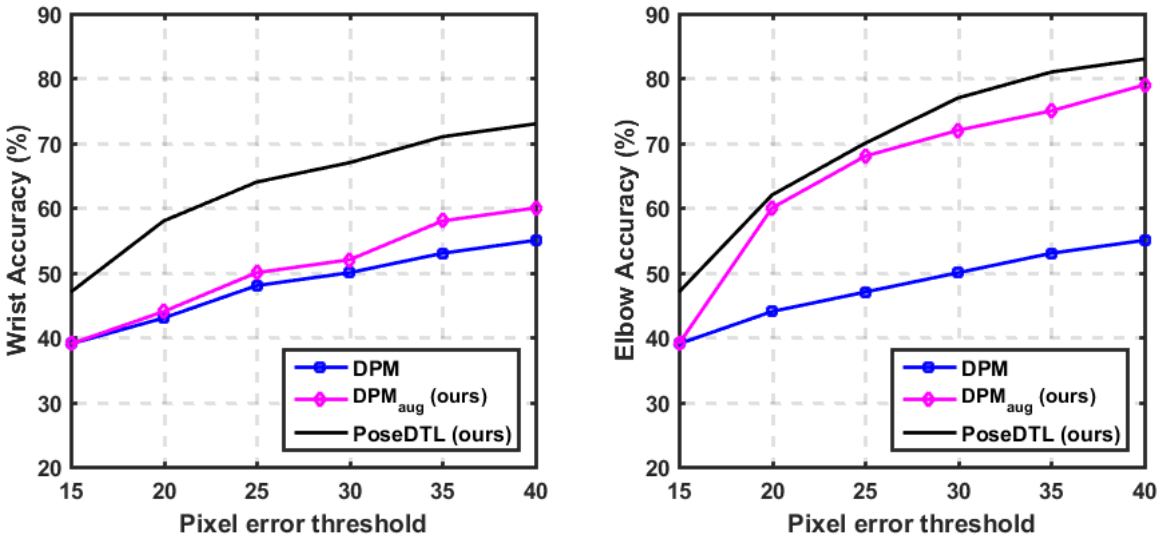


Figure 2.9: Pose in the wild dataset: The figures compare the DPM [141] results trained on FLIC dataset, without data augmentation, with our proposed data augmentation and by applying our proposed Pose-DTL approach.

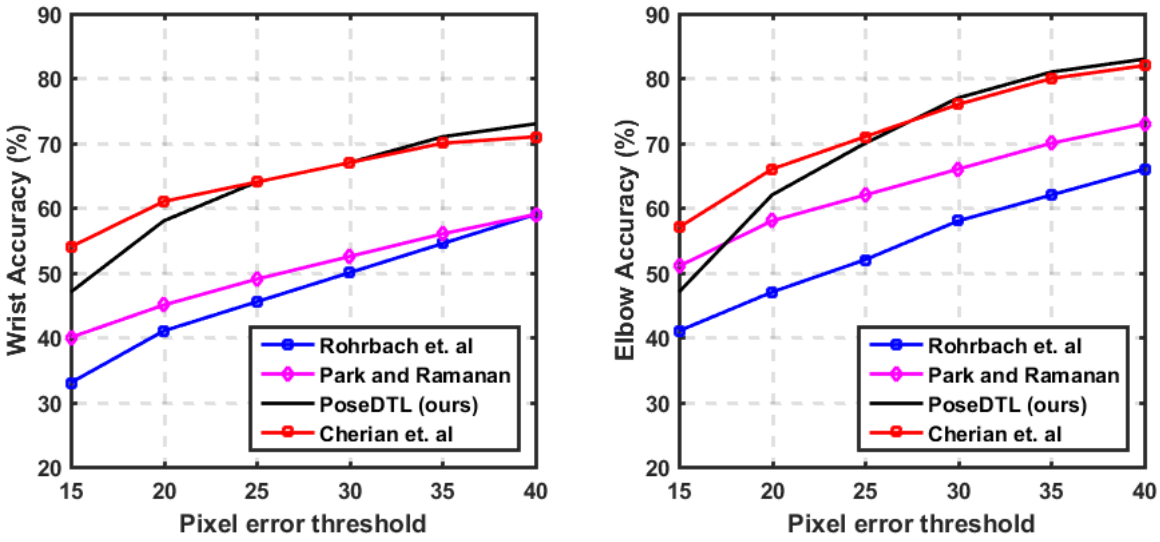


Figure 2.10: Pose in the wild dataset: The figures compare our pose-DTL approach with Rohrbach *et al.* [82], Park and Ramanan [68] and Cherian *et al.* [16] approaches.

Body part	r-wrist	l-writst	r-elbow	l-elbow	r-shoulder	l-shoulder
DPM	51.2	46.1	65.8	67.7	89.2	**93.3**
DPM+OF	55.8	48.5	67.4	68.7	89.7	93.3
Pose-DTL	**61.2**	**60.8**	**74.4**	**74.1**	**90.1**	92.8

Table 2.2: Results on the ADL Rochester pose dataset and comparison with the State-of-the-art results, ''l" represents left body parts and ''r" right body parts. As it is presented in the table, our model has better results than the other reported state-of-the-art results on this dataset, except for the left-shoulder

As it is presented in Fig 2.9, our data augmentation significantly improve the results particularly for the Elbows. Applying our model, improve the estimated pose by the evaluation on pixel distance of 40 by a large margine. For very small pixel distance, 15 pixels, we are ahead of the DPM baseline and our augmented DPM baseline, and better than some of the state-of-the-art approaches presented in Fig 2.10 but slightly lower than the reported results in [16].

ADL Rochester Pose dataset

Activities of Daily Living Rochester dataset [60] is mostly used for the daily living classification task. Rostamzadeh *et. al.* presents a significant improvement in the action classification results on this dataset by employing pose information [84].

Evaluation on Pose of the ADL Rochester dataset [84]

To make our results comparable with the state-of-the-art methods on this dataset, we rely on [84, 140]. As it is presented in the Table 2.2.3, results significantly improve for the wrists and slightly for the elbows. However, there is not much difference on the shoulders results and even for the left shoulder, we have slightly worse result. The reason is that, there is not any camera movement on the ADL dataset. So, most of the captured motion from the Dense Trajectories are useful motion and are for to the process of performing the action. So, the

results for the wrists and elbows which are actually involved on many actions and move improve significantly. However, shoulders are not much ocluded and even using DPM model [140], shoulders are well-localized. Optical Flow combined with DPM is shown to improve the results of wrists and elbows [84], and dense trajectory length (DTL) can be viewed as an extension of Optical Flow for the task of pose estimation in videos.

2.2.4 Conclusion

We presented a novel approach for human pose estimation in videos. Through the experiments, we show the length of the movements in consecutive frames can help localizing lower arm and upper arm body-parts. Moreover using an itterative approach and with a SIFT-inspired descriptor, we optimize the amount of motion information that should be embeded in wrists for any given frames.

Finaly using dynamic programing, we localize different body parts in frames of videos. We evaluated our method on two publically available datasets, (1) Pose in the wild and (2) ADL Rochester Pose datasets.

Chapter 3

Modelling Temporal Distribution of Features

In video retrieval, an important research problem is how to adequately capture temporal information. In this chapter[1], we address the problem of modeling the temporal variation by:

- Exploring the effect of Temporal Fisher Kernel model over frame-based Global Features [61].

- Capturing temporal variation by Temporal Fisher Kernel, Temporal VLAD and our proposed Hard and Soft Cluster Encoding models over local features [85].

Our first proposed solution is covered in 3.1 and the second one is covered in 3.2.

[1]This research is published in the following Computer Vision conferences: ACM Multimedia 2013 [61] and International Conference on Image Processing (ICIP) 2015 [85]

3.1 Time Matters! Capturing Variation in Time in Video using Fisher Kernels

3.1.1 Introduction

In video retrieval, an important research problem is how to adequately capture temporal information. Until recently, most video retrieval systems relied mostly on single representative video frames where time is ignored for efficiency reasons [107]. Recent work simply accumulates features over a whole video sequence [22, 40, 99]. Such accumulation may capture more information, but also mixes information, disregarding appearance variation over time. For example, when a car approaches and then turns a corner, there are first straight movements followed by turning movements. It is important that both types of movement do not happen at the same time. We want to have a representation which keeps this distinction.

In this research, we propose a novel video representation method which aggregates frame-based features while retaining their variation in time. Specifically, we propose to use the Fisher representation [69] which was recently introduced for improving a Bag of Local Visual Features approach [17]. Bag of Local Visual Features captures the visual variation in space for images and in both space and time for video. The Fisher Kernel improves over the common k-means vocabulary by modelling the distribution of features within each visual word. In contrast to Local Visual Features, in this work, we apply the Fisher representation on frame-based features, effectively capturing variation *in time only* (as there is no variation in space). Like Bag of Local Visual Features, all ordering is lost but all variation is captured. Using the Fisher representation for modelling variation in time, (1) dissimilar frames will be represented by different mixture components (i.e. clusters), preventing blending of unrelated features while enabling them to co-exist in a single representation. This enables representing videos which consist of dissimilar parts (which may not

even have a fixed temporal order) such as news broadcasts that switch between the news-anchor and on-site footage. Furthermore, (2) similar frames that fall in the same mixture component will be modelled with respect to the general distribution of that component, capturing subtle variations in time such as the different appearances of a person walking by.

We test our Fisher-based framework for modelling variation in time on a variety of video benchmarks: genre retrieval on the MediaEval benchmark [97], Human Action Recognition on the UCL50 dataset [78], and daily activity recognition on ADL [60]. Additionally, we employ a variety of frame-based features: global Histogram of Oriented Gradients (HoG) [18], global Histogram of Optical Flow (HoF) [78], global Colour Naming histogram (CN) [121], HoF-based body-part histogram [81], and even on block-based audio features [57]. We show that the Fisher representation consistently and significantly outperforms simple accumulation. Additionally, by explicitly modelling the variation in time we obtain state-of-the-art results or better on all three datasets using a smaller array of simpler features.

To summarise, our main contributions are the following: (1) We introduce a Fisher-based representation for frame-based features in video that captures variation in time. (2) We demonstrate its generality in terms of applications by applying it to genre-recognition, sports-recognition, and daily activity recognition. (3) We demonstrate its generality in terms of features by using audio features, global visual features, and body-part features. (4) We achieve similar or better performance than the state-of-the-art using a smaller array of simpler features.

3.1.2 Related Work

Recently, researchers have successfully captured local temporal information in video by using spatio-temporal features, visual features that are measured in the 3D volume spanned by the video frames. These features are extracted either at

interest points which are stable in both space and time [22, 24, 40, 45, 111], or at stable trajectories [128]. The specific movement pattern captured by these features yields significant improvements over 2D features. However, these features are accumulated over an entire video sequence ignoring the visual variation at different parts of the video.

Some works include some form of variation in time by using a linear quantization of the video: the video is split into n sequences of an equal number of frames, where for each sequence all features are accumulated [10, 45, 88, 128]. Histograms of the individual sequences are concatenated, leading to good accuracy improvements. In [108] the authors use a linear quantization method for global features, where the features are averaged inside a sequence.

Few works focus directly on modelling the temporal order/variation between frames [107]. There is some work on using Hidden Markov Models [42, 73]. Other work uses temporal rules with high-level concepts [52, 106].

3.1.3 Modelling Variation in Time

The Fisher Kernel [34] represents a signal as the gradient with respect to the probability density function that is a learned generative model of that signal. Recently, [69] introduced the Fisher Kernel as an improved visual vocabulary for Bag-of-Words. Its success shows that it meaningfully captures the visual variation of local descriptors.

In this work, we employ the Fisher Kernel to capture variation in time in video. We follow [69] and use a Gaussian Mixture Model with diagonal covariance matrices as generative distribution. Specifically, let μ_i and σ_i be the mean and standard deviation of the i-th Gaussian centroid, let $\gamma(i)$ be the soft assignment to the i-th Gaussian of the d-dimensional feature x_t captured at frame t.

The gradient of the GMM with respect to μ_i and σ_i are calculated as [69]:

$$\mathcal{G}^x_{\mu,i} = \frac{1}{T\sqrt{\omega_i}} \sum_{t=1}^{T} \gamma(i) \frac{x_t - \mu_i}{\sigma_i} \tag{3.1}$$

$$\mathcal{G}^x_{\sigma,i} = \frac{1}{T\sqrt{2\omega_i}} \sum_{t=1}^{T} \gamma(i) \left[\frac{(x_t - \mu_i)^2}{{\sigma_i}^2} - 1 \right]. \tag{3.2}$$

The final Fisher vector is the concatenation of the $\mathcal{G}^x_{\mu,i}$ and $\mathcal{G}^x_{\sigma,i}$ for $i = 1...k$ and has a dimensionality of $2kd$.

Interpreting the formulas in terms of variation in time, Equation 4.7 averages *related* features over time, which are related as they fall in the same mixture component. Equation 4.8 models the variation of related features over the video sequence, capturing subtle visual changes (e.g. a car driving by). The different mixture components capture drastic variations in time such as a shot changes.

Important parameters or design choices are: (1) Applying PCA on initial features x_t, reducing dimensions of the final Fisher vector and potentially improving GMM clustering by decorrelation. (2) The number of GMM clusters. (3) Normalization of the Fisher vector. (4) Choice of classifier.

3.1.4 Experiments

We demonstrate the advantages and generality of our framework on three different datasets using a variety of features. For brevity reasons we will mainly focus on the number of GMM clusters and only touch upon applying PCA. We normalise the Fisher vector by taking the square root followed by the $L2$-norm [69]. In contrast to [69], we use SVMs with RBF-kernels as these performed better than linear SVMs, even at an increased number of clusters for the latter. When combining different types of features we use weighted late fusion, learning weights on our optimization sets.

Genre Retrieval

We perform genre retrieval on the 2012 MediaEval Genre Tagging Task [97], consisting of 2000 hours in 14,838 videos, labelled according to 26 genres such as art, autos, and comedy. Performance is measured in terms of Mean Average Precision (MAP). We perform all parameter optimization on the training set which we split in two fixed, equally sized parts. We compare with the state-of-the-art using the official training set (5,288 videos) and test set (9,550 videos).

Baseline. We use the following features: (1) *Global Histogram of Oriented Gradients [18] (81 dimensions)* which calculates HoG over the whole frame using a 3x3 spatial division. (2) *Colour Naming histogram [121] (11 dimensions)* of the whole frame. (3) *Audio features [57] (98 dimensions)* which are general purpose audio descriptors extracted over a standard period of 1.28 seconds around the frame using [57]. Results of averaging features over the whole video are presented as the horizontal lines in Figure 3.1.

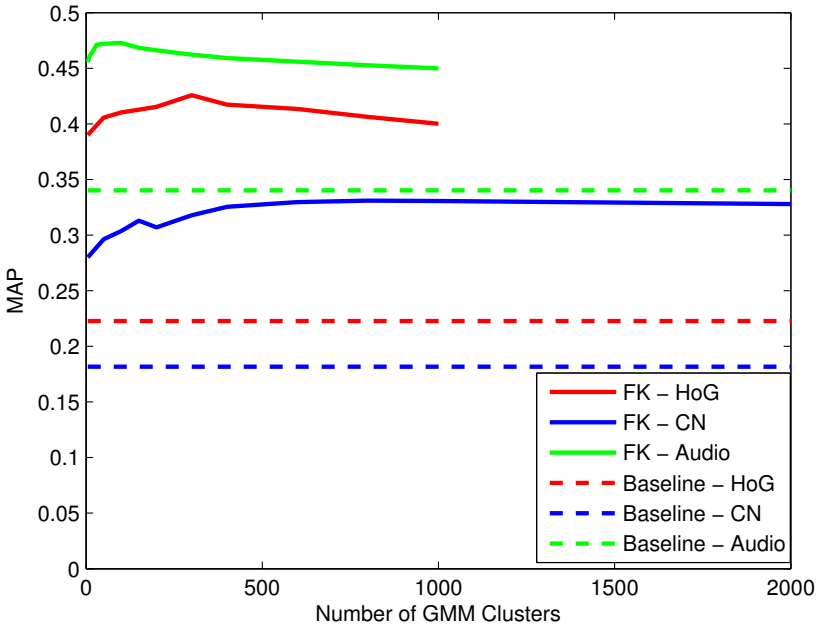


Figure 3.1: Mean Average Precision (MAP) while varying the number of cluster centres on the MediaEval 2012 training set.

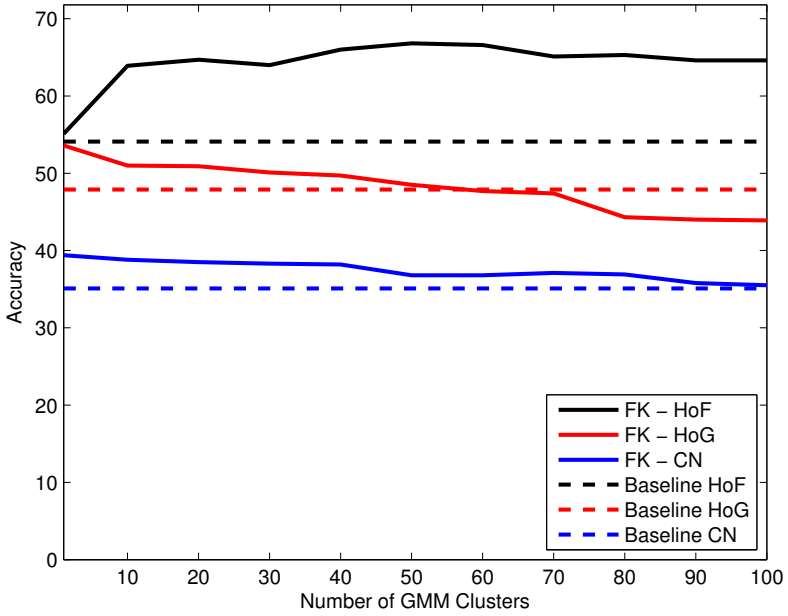


Figure 3.2: Classification accuracy on half of UCF50 sports while varying the number of cluster centres (8-fold cross-validation).

Optimizing the Fisher Representation. We ran experiments with PCA dimensionality reduction on the frame-based features, setting the number of cluster centres to 100. We found that for Colour Naming, applying PCA reduces performance. This is because the dimensions are decorrelated and non-redundant by design [121]. For HoG and Audio features the optimal reduction is to keep 80% of the dimensionality, where for HoG accuracy increased a full 5%. We choose these PCA settings for subsequent experiments.

Next, we determine the optimal number of clusters for each feature as shown in Figure 3.1. First of all, notice the big improvements of the Fisher representation over the baseline which simply averages the features: Even when using only a single centroid, Colour Naming goes up from 0.18 MAP to 0.28 MAP, HoG goes up from 0.22 MAP to 0.38 MAP, and Audio goes up from 0.34 MAP to 0.45 MAP. The modelling of variation in time therefore significantly improves results. Increasing the number of clusters increases performance even

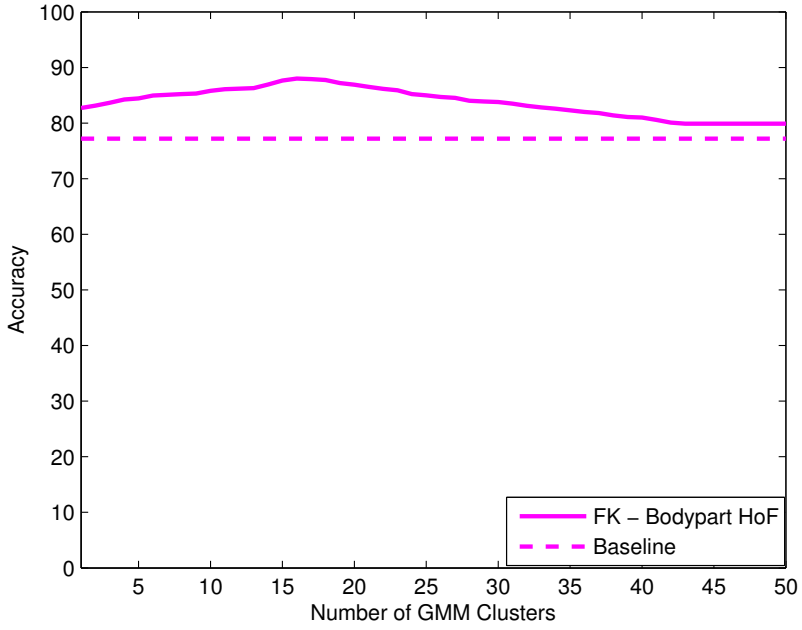


Figure 3.3: Classification accuracy on ADL daily activity recognition on half the dataset while varying the number of cluster centres.

further: Both Colour Naming and HoG increase an extra 0.05 MAP, reaching 0.33 MAP and 0.43 MAP at 800 clusters and 200 clusters respectively. Audio features increase to 0.47 MAP at 50 clusters. We will use this number of clusters in the next experiment. The final sizes of the Fisher vectors are reasonable at 17,600 for Colour Naming, 42,000 for HoG, and 9,000 for Audio features. Note that performance of both HoG and Audio features go down after the optimal points, likely due to the high dimensionality of the features (i.e. curse of dimensionality).

Comparison to State-of-the-Art. Results on MediaEval 2012 are shown in Table 3.1. For audio features our results are at 0.47 MAP much better than the best result of 0.19 reported at the MediaEval workshop [33]. For visual features only, at 0.46 MAP we perform significantly better than the best result of 0.35 MAP [100]. Remarkably, our combination of audio and visual features yields with 0.55 MAP a better performance than the use of text from automatic speech

recognition and meta-data, which had the highest performance at MediaEval 2012 at 0.53 MAP.

To conclude, using the Fisher kernel to model variation in time significantly improves over a simple averaging of features, yielding much better results than the state-of-the-art on the MediaEval 2012 benchmark.

Human Action Recognition

We now evaluate our framework on the UCF50 Human Action Recognition dataset [78], which contains 6600 realistic videos from Youtube with large variations in camera motion, object appearance and pose, illumination conditions, scale, etc. It has 50 mutual exclusive categories such as biking, diving, drumming and fencing. Performance is evaluated in terms of classification accuracy. We perform all optimization on half of the dataset, using 8-fold cross-validation. We compare with the state-of-the-art using the standard leave-one-group-out cross-validation on the full dataset [78].

Baseline. We use the following features: (1) *Global Histogram of Oriented Gradients [18] (9, 36, 81, and 144 dimensions)* which calculates HoG over the whole frame using a 1x1, 2x2, 3x3, and 4x4 spatial division. (2) *Global Histogram of Optical Flow [78] (9, 36, 81, and 144 dimensions)* which measures the average velocity of non-stationary pixels over a region in 9 orientations. We use a 1x1, 2x2, 3x3, and 4x4 spatial division. (3) *Colour Naming Histogram [121] (11, 44, 99, and 176 dimensions)* using a 1x1, 2x2, 3x3, and 4x4 spatial division. In all experiments, we combine different spatial divisions for a single feature type using late fusion with equal weights. Results of averaging each feature over the whole video are shown as horizontal lines in Figure 3.2.

Optimizing the Fisher Representation. We first optimized the dimension reduction using PCA. We found that both the Colour Naming histogram and the Histogram of Optical Flow did not benefit. For HoG we found a good improvement by reducing dimensions to 90% (data not shown).

Next, in Figure 3.2 we evaluate the performance with respect to the number of GMM clusters, where we use the same number of clusters for all spatial divisions of a single feature type. For Colour Naming and HoG the use of a single cluster improves the baseline with 6% and 5% respectively. More clusters degrade performance as for this dataset the visual changes are subtle and do not require different mixture components. For HoF, using 50 clusters improves the baseline of 54% to 67%, a 13% improvement. Hence the optical flow changes drastically in time which is best captured in multiple clusters. Indeed, for example a baseball pitch has at least three distinct movement patterns: static (before the action), the pitch, and the batting. In the next experiment we use 1 cluster for CN and HoG, and 50 clusters for HoF.

Comparison to State-of-the-Art. We present the state-of-the-art in Table 3.2. As can be seen, we rank second with 74.7% accuracy after the 76.9% accuracy of Reddy et al. [78]. However, we use only global features whereas all other good performing methods use computationally more expensive Space-Time Interest Points (STIPs). Only the GIST3D entry of [40] does not use STIPs. They use global, frame-based features plus linear quantization. Our performance using the Fisher vector is a significant 9.4% higher.

We conclude that our framework yields similar performance as the state-of-the-art while using simpler features.

Table 3.1: Comparison with State-of-the-Art (SoA) in terms of Mean Average Precision (MAP) on MediaEval 2012.

Feature type	**Summary SoA method MediaEval 2012**	**MAP SoA**	**MAP ours**
Audio	Block Based Audio Features and 5-NN [33]	0.192	0.475
Visual	Visual descriptors (Color, Texture, rgbSIFT) [100]	0.350	0.460
Audio & Visual	-	-	0.550
Metadata & Text ASR	BoW Text ASR & metadata [96]	0.523	-

Table 3.2: Comparison with State-of-the-art on UCF50 Human Action Recognition.

Method	Accuracy
Reddy et al. [78]	76.9%
This work	**74.7%**
Solmaz et al. [108]	73.7%
Everts et al. [24]	72.9%
Kliper-Gross et al. [40]	72.6%
Solmaz et al. [108]: GIST3D	65.3%

Table 3.3: Comparison with state-of-the-art on the ADL Daily Activity Recognition dataset.

Method	Accuracy
This work	**97.3%**
Wang et al. [128]	96.0%
Lin et al. [51]	95.0%
Messing et al. [60]	89.0%

Daily Activities

We report results on daily activity recognition using the ADL dataset [60], consisting of ten human activities such as dialling a phone, peeling banana, and chopping banana. Each activity is performed three times by five people, totalling 150 videos. Performance is measured in accuracy. We do all optimization on half of the dataset and report final results on the full dataset. In both cases we use leave-one-person-out cross-validation [60].

Baseline. As human pose and body-part motion are important for distinguishing the different categories, we extract body-part features [81]. We use the state-of-the-art body-part detector of [140] and extract at every frame for all 18 body-parts a Histogram of Optical Flow in 8 orientations (144 dimensions). The result of averaging this feature over the video is shown as the horizontal line in Figure 3.3.

Optimizing the Fisher Representation. We found no improvements by doing PCA on the bodypart HoF features.

Figure 3.3 shows accuracy with respect to the number of GMM clusters. Using only a single cluster yields a performance improvement from 77% to 82% accuracy. The best accuracy of 88% is obtained using 17 clusters. Note that the number of clusters is relatively low, likely due to the smaller dataset. At 17 clusters, the final feature has 4,896 dimensions. We use 17 clusters when testing on the full dataset.

Comparison to State-of-the-Art. We compare our work with others in Table 3.3. As can be seen, our approach yields the highest accuracy of 97.3%. This shows that the Fisher representation is also effective for modelling variation in time using local body-part features.

3.1.5 Conclusions

We propose to use the Fisher kernel to model variation in time for frame-based video features. While the temporal order is lost, the temporal variation is captured at two levels: similar features are grouped together while retaining variation, which enables capturing subtle variations over time such as a exhibited by a moving car. Dissimilar features are kept separate, preventing mixing features from unrelated parts of the video while keeping them in a single representation, which enables capturing different shots in a video.

We demonstrated that our framework is highly general: We showed significant improvements on a wide variety of features, ranging from global visual features, to body-part features, and to audio features. We also demonstrated that our method works on a wide variety of datasets: We obtained state-of-the-art performance on UCF50 using global features instead of the more complex STIPs used in other methods. We improved the state-of-the-art on ADL daily activity recognition. We significantly improved the state-of-the-art on the MediaEval 2012 genre classification task.

In future work we plan to model variation in time using Fisher kernels on more advanced features such as STIPs.

3.2 Cluster Encoding For Modeling Temporal Variation In Video

Videos change over time. They generally consists of different shots which vary wildly in appearance, while within a single shot the changes from frame to frame are more subtle. For automatic video classification, ideally such variation should be modeled. In this research, we propose a novel video representation in which we model the temporal variation in a video.

Currently, there are two main approaches for modeling video: (1) Bag-of-Words models [45, 127, 119] sample spatio-temporal video patches at specific locations in the video, from which local appearance or motion descriptors are extracted. Then Bag-of-Words techniques such as the Fisher Kernel [70] are used to model the variation of these descriptors, where temporal and spatial variation is indiscriminately mixed together. This seems suboptimal since spatial and temporal dimensions are fundamentally different; (2) Some works model the temporal *order* within a video by using Hidden Markov Models [42, 74]. However, such models are generally slow at both training and testing time.

In this work, we take another approach. Unlike Bag-of-Words, we want to explicitly model the variation in time. However, instead of modeling temporal *order*, we only model the temporal *variation*. In particular, we first create frame-based features which model the appearance or motion at a specific point in time. Afterwards, we model the variation of these frame-based features, which means the temporal order is lost but we explicitly model the variation in time. The resulting representation is richer than the classical Bag-of-Word approach, while at the same time resulting in a representation which is fast to create and easy to use. We demonstrate the benefits of our framework using visual features on the ADL Rochester dataset [60] and on Blip10k [98].

The approach of modelling only the temporal variation was earlier proposed in [61]. In this work, we improve their work in two important ways: (1)

Whereas [61] used global frame features (i.e., one frame is described by a single large HOG/HOF descriptor), we use the more powerful Bag-of-Word features to represent a single frame; (2) Furthermore, we introduce two novel temporal aggregation techniques which are effective in modeling the variation of the frame-based Bag-of-Word features.

3.2.1 Related work

The current dominant method of creating video features is the Bag-of-Words method [103, 17]. This approach samples local spatial-temporal video patches on either space-time interest points [22, 45], a regular grid [119, 129], or dense trajectories [127]. From these patches local descriptors are extracted such as Histograms of Oriented Gradients (HOG) [18], Histograms of Optical Flow (HOF) [19, 45], and Motion Boundary Histograms (MBH) [19]. The set of descriptors is subsequently modeled by counting codewords of a visual vocabulary (e.g., [103, 17, 122]), the Fisher Kernel [70], or VLAD [37]. These approaches all model the variation of the descriptors, but make no distinction between variation in time and variation in space. For videos consisting of multiple, wildly different shots, this is likely suboptimal. In this work, we use these classical techniques to create frame-based Bag-of-Word features. However, afterwards we perform a separate aggregation step to model the variation of these frame-based features in time. This means our representation explicitly models temporal variation.

Several approaches explicitly model the temporal *order* of frames within a video by using Hidden Markov Models [105, 42, 74]. Revaud, et al. [79] encode frame descriptors jointly in the frequency domain while keeping the temporal order. Other works employ temporal rules with high-level concepts [53]. These works are usually time consuming, since a pre-temporal-segmentation or a temporal constraint is required to be applied. In this work, we propose to drop the temporal *order* but keep the temporal *variation*. This leads to a simpler yet

faster architecture with the same or better performance.

3.2.2 Method

We create video representations in a two-step sequence: (1) we use a standard Bag-of-Words method to create frame-based features. These features model the spatial variation of local descriptors in the video at a specific time; (2) we use standard and new aggregation methods to model the variation of frame-based features in time.

Creating Frame-based Visual Features

We use standard methods to create frame-based visual features. As local visual descriptors we extract densely sampled HOG and HOF features using the software[2] and recommended settings of [119].

We experiment with three types of aggregation methods: hierarchical k-means (HKmeans) plus codebook assignment (e.g., [17, 119]), the Fisher Kernel [70], and VLAD [36], for which we use the VLfeat toolbox[3] [123]. Importantly, we use these methods to create *frame-based* features. That is, in contrast to normal approaches which use Bag-of-Words to represent all descriptors in the video, we create a representation at each point in time for which we have descriptors[4].

For all Bag-of-Words methods we use the recommended settings. We normalize our frame-based Bag-of-Words representations for HKmeans using the the square root followed by L_1. For VLAD and FK we apply the square root while keeping the sign followed by L_2.

[2] http://huppelen.nl/publications/RealtimeHofHogReleaseV1.0.zip

[3] http://www.vlfeat.org

[4] Local descriptors actually span multiple frames but have a time-stamp in the middle of these frames. We only aggregate features with equivalent time-stamps. So the term "frame-based" features is not 100% accurate but it captures the spirit of our work the best.

For future reference, let us represent a set of N videos as $\{V_1, V_2, ..., V_N\}$. We denote the number of frame-based features in a video V_j by η_j. For the m^{th} layer ($m \in \{1, ..., \eta_j\}$) of video V_j, $\phi_{j,m}$ represents the frame-based vocabulary assignment.

Temporal Encoding of Frame-based features

We want to explicitly capture temporal variation over the frames within a video. We use two classical methods to encode the temporal variation over frame-based features, the Fisher Kernel and VLAD. Then we propose a novel method inspired by these models that outperforms both.

Temporal Fisher Kernel Encoding (TFK). We use the Fisher Kernel [70] to encode the temporal variation over frame-based features. This means that for each video V_j all frame-based features $\phi_j = \bigcup_{m=1}^{\eta_j}\{\phi_{j,m}\}$, are assigned to N_c clusters with a Gaussian Mixture Model (GMM).

Let μ_i and σ_i be the mean and the standard deviation of the i^{th} Gaussian component, $\gamma(i)$ the soft assignment of $\phi_{j,m}$ to Gaussian i, and ω_i the mixture weight of Gaussian i. Let D denote the dimensionality of ϕ_j. Now $G_{\mu,i}^{\phi_j}$ and $G_{\sigma,i}^{\phi_j}$ are respectively the D-dimensional gradient for the mean (μ_i) and standard deviation (σ_i) of Gaussian i. Mathematical derivations [70] lead to:

$$G_{\mu,i}^{\phi_j} = \frac{1}{\eta_j\sqrt{\omega_i}} \sum_{m=1}^{\eta_j} \gamma_m(i) \frac{\phi_{j,m} - \mu_i}{\sigma_i} \tag{3.3}$$

$$G_{\sigma,i}^{\phi_j} = \frac{1}{\eta_j\sqrt{2\omega_i}} \sum_{m=1}^{\eta_j} \gamma_m(i) \left[\frac{(\phi_{j,m} - \mu_i)^2}{{\sigma_i}^2} - 1\right] \tag{3.4}$$

where the divisions between vectors is a term-by-term operation. The final gradient vector G^{ϕ_j} is the concatenation of the $G_{\mu,i}^{\phi_j}$ and $G_{\sigma,i}^{\phi_j}$ vectors, for $i = 1, \ldots, N$ and has a dimensionality of $2DN$. We normalize this vector by taking the square root while keeping the sign, followed by L_2, as recommended by [70].

Temporal VLAD Encoding (TVLAD). VLAD encoding [36] is introduced as a simplified alternative to FK.

In our VLAD temporal encoding, the frame-based features $\phi_j = \bigcup_{m=1}^{\eta_j}\{\phi_{j,m}\}$ for video V_j are assigned to $N_c = \{c_1, c_2, ..., c_n\}$ vocabularies that are obtained by using *K-means*. Features are assigned only to the nearest cluster centre: $\kappa_m(i) = 1$ if the feature belongs to cluster m, and 0 otherwise. This yields:

$$R_{\mu,i}^{\phi_j} = \sum_{m=1}^{\eta_j} \kappa_m(i) \left[\phi_{j,m} - \mu_i\right] \tag{3.5}$$

The final VLAD vector R^{ϕ_j} is the concatenation of $R_{\mu,i}^{\phi_j}$ for $i = 1, \ldots, N$ and has dimensionality DN. We normalize this vector by taking the square root while keeping the sign, followed by L_2, as recommended by [36].

Temporal Hard Cluster Encoding (THC). The VLAD and Fisher Kernel both alter the feature space with respect to the cluster centres. While this has proven to work well on local descriptors, it is unclear if that is optimal. When our frame-based features are created using a codebook assignment, the resulting features are histograms. For such histograms, measuring distances using Histogram Intersection (or, equivalently, Manhattan distance) is natural and has proven to work well (e.g., [119]). Hence, we propose an alternative encoding which does not alter the original feature space, but which accumulates the features within each cluster. As in VLAD, we use *K-means* to create a visual vocabulary and let $\kappa_m(i)$ denote the hard assignment to a cluster. Let N_m denote the number of assigned features to cluster m. Now we model the average *of the features within a cluster* as:

$$S_{\mu,i}^{\phi_j} = \frac{1}{N_m} \sum_{m=1}^{\eta_j} \kappa_m(i)\phi_{j,m}. \tag{3.6}$$

We model the standard deviation of the features within a cluster as:

$$S^{\phi_j}_{\sigma_m,i} = \frac{1}{N_m} \sum_{m=1}^{\eta_j} \kappa_m(i) \left[(\phi_{j,m} - S^{\phi_j}_{\mu,i})^2 \right]. \tag{3.7}$$

As before, the final representation is obtained by concatenation. Note that unlike VLAD and Fisher, these formulas do not use the location of the cluster centre of the original k-means clustering. We normalize using the L_1 norm, as this preserves frame-based features which are based on codebook counts.

Temporal Soft Cluster Encoding (TSC). The assignment of a feature to a single cluster may be quite crude, especially when there are few clusters. Hence, we also propose a soft assignment version of our encoding scheme, as it is also done in the Fisher Kernel and in [122]. For this soft assignment $\gamma_m(i)$ we assume that all clusters created by the k-means clustering algorithm have an isotropic covariance $\sigma = \lambda I_D$, where λ is a parameter we optimize and I_D is the D-dimensional identity matrix. This leads to:

$$T^{\phi_j}_{\mu,i} = \frac{1}{N_m} \sum_{m=1}^{\eta_j} \gamma_m(i) \phi_{j,m} \tag{3.8}$$

$$T^{\phi_j}_{\sigma_m,i} = \frac{1}{N_m} \sum_{m=1}^{\eta_j} \gamma_m(i) \left[(\phi_{j,m} - T^{\phi_j}_{\mu,i})^2 \right] \tag{3.9}$$

As before, we normalize using the L_1 norm.

3.2.3 Experimental Setup

We evaluate our temporal encoding schemes on the Rochester ADL [60] and Blip10k [98] datasets. We first describe these datasets and then give details on our experimental setup.

Datasets

Rochester Activities of the Daily Living (ADL) dataset. This dataset consists of ten complex fine-grained human activities. Each activity is performed three

times by five different people, with different ethnicity, appearance and manner of performing the actions. Each clip is in the range of 3-50s. In total the dataset contains 150 videos.

Blip10k dataset. Blip10k contains videos from Blip.tv [98]. The dataset contains 14832 episodes with the running time of 3288 hours. Each video is labeled according to 26 web specific video genre categories, e.g., art, autos and vehicles, business, comedy, etc. The dataset was successfully validated during the 2010-2012 MediaEval benchmarking campaigns.

Baseline. Our *baseline* uses the simplest temporal encoding of frame-based features. We aggregate by taking the mean and standard deviation. So basically, this models the temporal variation by a single Gaussian distribution. Note that this corresponds to taking a single cluster for our THC and TSC encodings.

Temporal encodings. We model the distribution of our frame-based features using 5 different temporal encodings: (i) *Hard Cluster Encoding (THC)* (ii) *Soft Cluster Encoding (TSC)* (iii) *Fisher Kernel Encoding (TFK)* [70] (iv) *VLAD Encoding* with the $kmeans$-based vocabulary *(TVLAD-K)* [36] and (v) *VLAD Encoding* with the $GMMs$-based vocabulary *(TVLAD-G)* [71].

Optimization of λ. For the TSC encoding, λ is chosen from $\lambda_{set} = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ to preserve the softness of the representation using an inner-loop cross-validation over the training data. For Blip10k and Rochester ADL we found the optimal values to be respectively $\lambda = 10^{-2}$ and $\lambda = 10^{-4}$.

How many clusters for temporal encoding. This is one of the main questions of this work. We vary the number of clusters from 1 to 5. Preliminary experiments showed no significant improvements for more clusters, while more clusters significantly increase the dimensionality of our video representations (it scales linearly, but initial dimensionality is already substantial).

3.2.4 Results and Discussions

Rochester ADL dataset [60]. We first perform our evaluation on the ADL

dataset using leave-one-person-out cross-validation, a standard procedure for this dataset. We have evaluated all the possible combinations of frame-based features, temporal encoding approaches and SVM kernels (Linear, RBF, Histogram Intersection), but given the space limitation we only present the best and most relevant results. Figure 3.4(a) shows a comparison between several temporal encoding methods, where our frame-based features are HOF+HKmeans. Note for the SVM kernel, we selected for each method its best kernel: a linear kernel for VLAD and FK, corresponding to recommendations of [70, 36]; the Histogram Intersection kernel for THC and TSC as anticipated in Section 3.2.2.

First of all, we observe that all temporal encodings benefit from having multiple clusters. Intuitively, this means that the visual variation in the videos over time is too high to be captured by a single cluster only. However, the classical VLAD and Fisher Kernel encodings do not work well on the frame-based Bag-of-Words features: these are worse than the baseline which models the frame-based features by a single Gaussian distribution. Still, our novel Soft Cluster Encoding yields significant improvements: accuracy goes up from 88.0% to 94.6%, an improvement of 6.6 percentage points.

In the next experiment, we keep the best temporal encoding method, i.e., Soft Cluster Encoding, but instead change the frame-based features. Here we create frame-based features not only using HKMeans encoding, but also using VLAD and Fisher Kernels. The results in Figure 3.4(b) show that for all frame-based features the temporal encoding improves the accuracy by 3-7%. We can also note that HKmeans are the best frame-based features.

We also compare our results with the state-of-the-art in Table 3.4. The best results are obtained by methods which use complex features such as Body-Parts [61, 84] or by a method which models contextual interaction between interest points [132]. However, if we compare our results to approaches relying only on fast local visual descriptors, we obtain significantly better results: the best method [4] has 88.6% using both HoG and HoF, similar to our baseline. In

Table 3.4: Accuracy results for different approaches stating explicitly the used features. For the Blip10k dataset, V denotes visual features. The results obtained by our approaches are depicted in bold.

Rochester ADL dataset	
Features	Acc.
Our method-HoF	**94.6%**
Our Baseline-HoF	**88.0%**
HoF+HoG [4]	88.6%
HoF+HoG [132]	85.0%
HoF [94]	80.0%
HoF+FG+PoseDets [84]	98.8%
HoF+PoseDets[61]	97.3%
HoF+HoG+ContextFtrs [132]	96.0%
KPT+color+FaceDets [60]	89.0%
HoF+HoG+KPT [77]	82.6%
Blip10k dataset	
Features	Acc.
Our method-HoG (V)	**49.6%**
Our Baseline-HoG (V)	**42.2%**
G-HoG+Color (V) [61]	46.0%
Color+RgbSIFT (V)[98][5]	35.0%
Audio+Video [61]	55.0%
Audio [61]	47.5%
Audio [98][4]	19.2%

contrast, we obtain 94.6% accuracy using only HoF. This shows that our explicit coding of temporal variation is very effective.

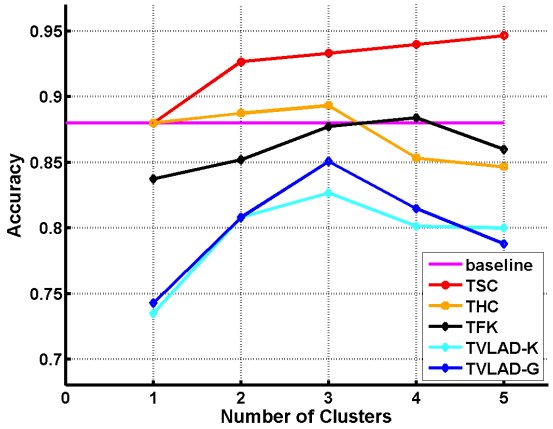
Blip10k [98]. Since the Blip10k dataset is huge, we only evaluate the framework that gave the best results on the Rochester ADL dataset. We replaced however the HoF descriptors with HoG descriptors to reduce the computation time. Consequently, as frame-based features we have a BoWs representation using HoG descriptors modeled by a HKmeans codebook. We model the temporal variation using Soft Cluster Encoding.

The results are presented in Figure 3.4(c). As before, the results go up drastically by properly modeling the temporal variation. Results improve from 42.2% to 49.6%, an increase of 7.4 percentage points in accuracy.

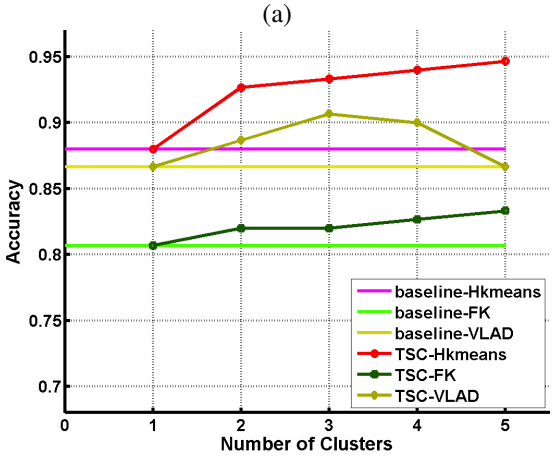
Table 3.4 shows the comparison with the state-of-the-art. Here, the best results are obtained by [61] while using a combination of visual and audio features. However, for visual features only, we outperform [61] by 3.6%, even if we use fewer visual features. We conclude that our explicit modeling of temporal variation in video is very effective.

3.2.5 Conclusion

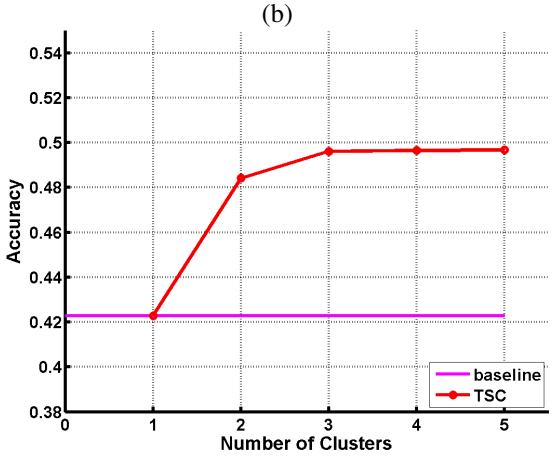
We presented a framework in which we explicitly model variation in time in video. First we create frame-based features based on Bag-of-Words. For modeling their variation in time (but not their order) we introduced Hard and Soft Cluster Encoding, novel encoding techniques inspired by the Fisher Kernel and VLAD. Results show significant improvements of respectively 6.6% and 7.4% accuracy over our baseline. Furthermore, comparing our results on the Rochester ADL dataset to other works which use only local visual HoG and HoF descriptors, we show accuracy improvements of 6%. On Blip10k, we outperform the state-of-the-art when using visual features only by 3.6%.



(a)



(b)



(c)

Figure 3.4: Experiments on the Rochester ADL dataset: (a) the performance of different encoding approaches with a fixed *BoW* extraction method; (b) the performance when using a fixed encoding method (TSC) and different frame-based representations. The performance using the best pipeline on the Blip10k dataset is shown in (c). All the graphs are shown when the vocabulary size changes from 1 (no temporal variation) to 5 (highest temporal variation).

Chapter 4

Daily Living Activities Analysis in Videos

Motivated by applications in areas such as patient monitoring, telerehabilitation and ambient assisted living, analyzing activities of daily living is an active research topic in computer vision and image processing.

In this chapter[1], we address the problem of Activities of Daily Living:

- In section 4.1, first, at feature level, we propose a method to extract and combine low- and high-level information and we show that the performance of body pose estimation (and consequently of activity recognition) can be significantly improved. Then motivated by the results that we obtained, for modeling the temporal variation, we apply Temporal fisher kernel to model the Temporal distribution of our new feature representation. [84].

- In section 4.2, we address the problem of analyzing unlabled data by proposing a Multi-Task Learning approach. The approach is motivated by this assumption that in daily living activities, people usually perform similar sets of actions in the same environment.

[1]This research is published in the following Computer Vision conferences: International Conference on Image Analysis and Processing (ICIAP) 2013 [84] and International Conference on Image Processing (ICIP) 2014 [138]

4.1 Daily Living Activities Recognition via Efficient High and Low Level Cues Combination and Fisher Kernel Representation

Automatic video scene understanding and activity analysis are active research topics in computer vision. In this work we focus on daily living activity scenarios. The interest in activity recognition in this scenario is motivated by the promise of important applications in areas such as patient monitoring and ambient assisted living.

Analyzing daily living scenarios is a challenging task. First of all, in such a scenario, different activities differ only slightly in motion and appearance. In some cases, the differences in appearance of the subjects performing the same task are more evident than the difference in activities. Moreover, one activity can be performed in many different ways, while two different activities may be performed in a very similar manner with respect to motion and appearance. For example, dialing and answering the phone are activities only slightly different in terms of hand movements. Particularly, if we consider the Activity of Daily Living (ADL) dataset[2], the difference between two activities in most cases is limited to taking *phone, banana* or *knife* from the *table, shelf*, or *refrigerator* and doing slightly different other activities (*e.g. eat snack* and *drink water*).

Recently, Bag-of-Words (BoW) models relying on local features have become popular in dynamic scene understanding due to their robustness to noise and occlusions. However, the traditional BoW representation that has typically been applied in activity recognition scenarios has some substantial restrictions [7, 46, 136]. First of all, a BoW representation based on low-level cues limits the access to the high-level information that may be more discriminative. Secondly, being a frequency histogram of quantized local appearances or motion, the relationships between temporal cues are totally ignored.

[2] www.cs.rochester.edu/∼rmessing/uradl/

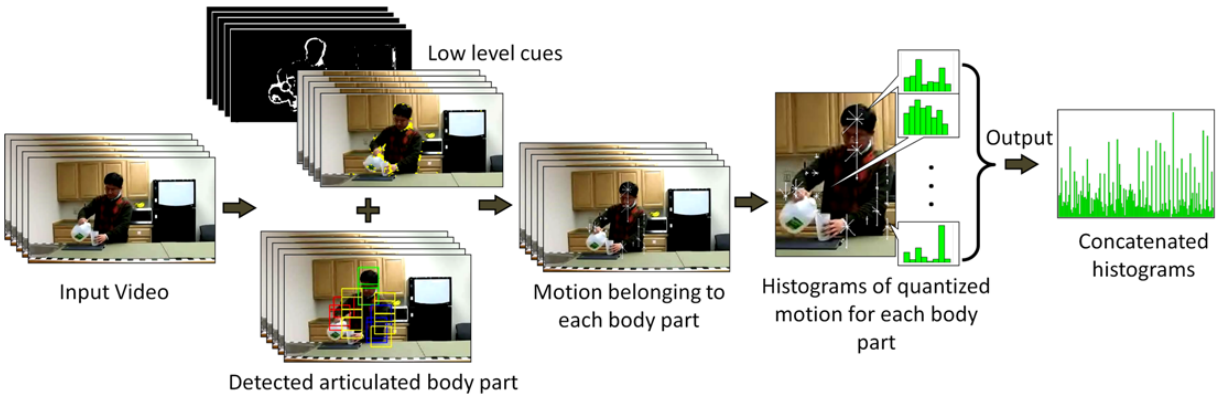


Figure 4.1: An overview of the pipeline we propose for human action recognition.

In this work we address these two drawbacks in the BoW representation. First, we make an enriched descriptor by combining low- and high-level cues that are obtained from the local motion and a body pose detector. In this way, the source of motion (*e.g.* a hand) is taken into account. Second, we apply a Fisher Kernel representation of the combined low- and high-level features to model the temporal variation. Additionally, we provide an efficient method for body pose estimation which builds upon [141] and allows us to improve the detector performance on a new dataset by simply exploiting the information provided by easy-to-extract low level cues (thus saving the cost of creating the ground-truth and re-training the detector). Finally, we apply the popular non-linear SVM classification method and show that the obtained results outperform the state-of-the-art on the ADL dataset.

4.1.1 Related Work

Typical approaches for activity recognition rely on a two-steps paradigm. The first step concerns the generation of feature vectors: features are extracted and quantized according to a pre-defined codebook and are accumulated to form the so called bag-of-words. The second step takes these bags-of-words as input and learns how to classify the different actions. This phase is generally supervised

and a training set is available for learning.

The first step is crucial for the good performance of the second one. In fact, the information discarded at this step can hardly be recovered afterwards. For example, if the codebook is defined based on local motion (*e.g.* tracklets or optical flow), all the information about the structure of the scene or about the entity involved in the motion is discarded. This causes a huge information loss, which heavily limits the capability of comprehending a scene in the learning step that follows. Over the past years, many works addressed this limitation and much effort has been devoted to enrich the descriptors with additional information beyond motion: (i) some works take into account the relationship between the spatio-temporal local features [30, 41, 58, 95]. Zhang *et al.* [148] enriched their descriptor not only with the relationship between neighboring local space-time features but also by considering the long-range relationship of local features. (ii) Malgireddy *et al.* [56] and Kovashka *et al.* [41] combined local features and made enriched descriptors. Others proposed taking the contextual features of interest points into account in a BoW representation [6, 131]. (iii) Lately, an increasing number of works exploited the information coming from detectors as a high level information about the observed scene [60, 82, 89, 146]. This is a step towards a higher level comprehension of the scene, w.r.t considering only low- or mid-level information represented by the local motion (*e.g.* optical flow, tracklets) or the local appearance (*e.g.* SIFT, HOG). In this way, the nature of the body parts involved in the observed motion is considered. In our daily living scenario, the person is monitored from a camera in a controlled environment and the body is clearly visible and mostly not occluded. We combine local motion with the high-level information coming from a body limbs detector. To do so, an efficient and accurate body pose estimator is required.

Over the past decade, many approaches have been proposed for capturing human body parts [27, 28, 39, 75, 116]. These works focused on generalizing and extending the pictorial model. Using a pictorial structure as a model

to represent human body pose is a popular approach that tries to model an object by its parts arranged in a deformable configuration. The problems of the variety of body part appearances, different orientations, and different scales in which humans may appear were not well-investigated in the traditional pictorial structure. Felzenszwalb *et al.* [26] proposed an extension of the pictorial model to detect objects at different scales using a multi-scale HOG-pyramid. Yang and Ramanan [141] proposed a more general pictorial model covering a variety of body configurations. Their proposed approach is among the most efficient works that model the human body skeleton as a tree. They detect small bounding boxes around the body parts instead of complete body limbs. This makes their work more efficient because it prevents the problem of double counting. In their work, a local appearance template is obtained by a multi-scale HOG descriptor [20] that allows detection at different scales. Our human pose estimator is built upon their work [141].

Finally we investigate the use of the Fisher Kernel representation to model the temporal variation of videos. Involving the temporal variation is not very well-investigated yet [105]. Kuehne *et al.* [43] and Qi *et al.* [74] used Hidden Markov Models. Other works employed temporal rules with high-level concepts [53]. To the best of our knowledge the only work that used Fisher Kernel to model the temporal variation in videos is [62]. They employed a frame-based global feature descriptor for a movie-genre classification scenario. In our work, we use Fisher Kernel to model the temporal variation over local descriptors of individual body-parts that are detected in consequent frames of a video in an action recognition scenario.

4.1.2 Our Method

We propose a novel activity recognition method obtained by combining information taken from both the local motion and the body part detector. Combining low- and high-level cues exploits the advantages of both cues: on one side the

robustness of low-level cues (*e.g.* optical flow) *w.r.t* occlusions, on the other side having the information about the body part involved in an activity increases the scene disambiguation.

In the case of body pose estimation, a significant drop in accuracy has been observed when a detector is trained on one dataset and it is evaluated on a different one [82]. The reason is that for some cases there are not enough samples in the training set. As the detector gives more priority to the positive samples of training set, the chance of detecting uncommon (w.r.t positive samples) body poses decreases. A possible solution to this is to obtain the body pose groundtruth for the new dataset and re-train the classifier. However, this procedure is very expensive and requires a considerable delay every time a new dataset has to be analyzed. Instead of training another classifier on the new dataset, we propose to use the already trained classifier, but we provide some additional information from the new dataset. Specifically, we used the classifier trained on the Buffy dataset [28], using the approach from [141]. Then we boost the classifier by exploiting the information of low-level cues from the ADL dataset. These low-level cues (*i.e.* optical flow and foreground pixels) can be easily extracted from a stationary webcam as in our case. To evaluate our contribution for pose estimation in a new dataset, we annotated the upper body poses for 371 frames obtained from different clips of the ADL dataset[3]. Then, we create our descriptors by combining our enhanced body-pose estimator with the local motion (*i.e.*, optical flow), that is already extracted for enhancing the pose estimator. Finally, we apply a Fisher Kernel representation to our descriptors to model the temporal variation in video, and apply a popular non-linear SVM classifier (SVM with RBF kernel) on our descriptors to classify the activities. The details of our approach are provided in the following sections.

[3]The groundtruth is available at: https://sites.google.com/site/negarrostamzadeh/Ground-Truth.7z

Body Pose Estimation

Pictorial structures model the body as an ideal template represented as a graph, *G=(V,E)*, in which single body parts templates (V) are connected with springs (E) that represent the geometric constraints between them. The placement of these springs can change, while the structure of the model is preserved. These deformations present different possible configurations of body parts. Each possible body configuration is given a score that is based on the sum of local and pairwise scores [26, 27]:

$$S(I,p,t) = \sum_{i \in V} w_i^{t_i} \phi(I,p_i) + \sum_{i,j \in E} w_{ij}^{t_i,t_j} \psi(p_i - p_j) + S(t) \tag{4.1}$$

where $\phi(I,p_i)$ is a HoG descriptor extracted from the pixel location p_i in image I and $\psi(p_i - p_j)$ is the relative location of part i with respect to part j. The first term in Eq 4.1 represents the *local* score (also called *appearance model*) that indicates how likely is that a template $\omega_i^{t_i}$ for part $i \in \{1,...,K\}$ of the body, tuned for type t_i, is located at position $p_i = (x,y)$ in the image I. The second term represents the *pairwise* score (also called *deformation model*) and controls the relative location of part i with respect to j. $S(t)$ is a *compatibility function* defined as,

$$S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{i,j \in E} b_{ij}^{t_i,t_j} \tag{4.2}$$

where $b_i^{t_i}$ represents the bias that favors particular type assignment for single part i and $b_{ij}^{t_i,t_j}$ represents the pairwise co-occurrence of parts i and j.

Our work builds upon [141] where the body relational graph is as a tree. The inference corresponds to maximizing the score function $S(I,p,t)$ over p and t and it can be efficiently solved with dynamic programming when the relational graph $G = (V,E)$ is modeled as a tree:

$$S_i(t_i,p_i) = b_i^{t_i} + w_{t_i}^i \phi(I,p_i) + \sum_{k \in kids(i)} m_k(t_i,p_i) \tag{4.3}$$

where $m_k(t_i,p_i)$ collects the message from the children of part i (located at p_i for the type t_i). In Yang *et al* [141], the local score (the second term in Eq. 4.3) is based only on the appearance cues (*i.e.* HOG). Differently from them, in our work, we use a model that is trained on a dataset (Buffy dataset [28]) and we enrich the local score by including information provided by the local cues, such as *foreground* and *optical flow*, calculated for a new dataset (ADL dataset):

$$S(t_i,p_i) = b_i^{t_i} + w_{t_i}^i \phi(I,p_i) + \alpha\beta S_{FG}^i(p_i,\gamma) + (1-\alpha)\eta S_{OF}^i(p_i,\lambda) + \sum_{k \in kids(i)} m_k(t_i,p_i) \tag{4.4}$$

In Eq. 4.4, local *foreground* and *optical-flow* information are combined with the local appearance information at the testing level. S_{FG} and S_{OF} respectively present foreground and optical flow scores corresponding to the information that comes from these local cues. In our representation the impact of S_{FG} and S_{OF} is controlled respectively by parameters β and η. Moreover, the relative impact of the two added terms is controlled by the parameter α.

Computing the foreground score (S_{FG}). The foreground score S_{FG}^i is defined as the percentage of foreground pixels contained in the corresponding body part's bounding box, centered at location $p_i = (x,y)$. In order to extract foreground pixels, we applied the dynamic Gaussian Mixture background subtraction model [110]. For the foreground score, we consider a smaller bounding box w.r.t the one used for computing the HOG features, otherwise we would include some unnecessary portion of the background. In particular, we compute the number of foreground pixels $|pixels_{FG}^{\{p_i,\gamma\}}|$ in a bounding box of size $L_{FG} = \frac{1}{\gamma}L$, centered at p_i, where L is the size of the appearance bounding box. In the experimental section we report the effect of varying γ.

The foreground score S_{FG} is computed as follows:

$$S^i_{FG}\left(p_i, \gamma\right) = \frac{|pixels^{\{p_i,\gamma\}}_{FG}|}{|pixels^{\{p_i,\gamma\}}|} \tag{4.5}$$

where $|pixels^{\{pi,\gamma\}}_{FG}|$ represents the number of foreground pixels that are present in a box centered at p_i with size L_{FG}, and $|pixels^{\{p_i,\gamma\}}|$ represents the total number of pixels in the foreground bounding box.

Computing the optical flow score (S_{OF}). We use the Lucas-Kanade optical flow algorithm [115]. Similarly to the foreground score, we compute the number of optical flows $|pixels^{\{p_i,\lambda\}}_{OF}|$ in a bounding box of size $L_{OF} = \frac{1}{\lambda}L$, centered at p_i. The optical flow score is formulated as follows:

$$S^i_{OF}\left(p_i, \lambda\right) = \frac{|pixels^{\{p_i,\lambda\}}_{OF}|}{|pixels^{\{p_i,\lambda\}}|} \tag{4.6}$$

where $|pixels^{\{p_i,\lambda\}}|$ represents the number of pixels in the optical flow bounding box.

Activity Recognition

For the low-level cues, we quantize the motion vectors into 8 possible directions. For the high-level cues we apply our enhanced pose estimator and detect the placement of N_{bp} body-parts (in this experiment $N_{bp} = 18$). Then we make an 8 bin histogram for each body-part. Optical flows are assigned to the corresponding body part. Finally, we concatenate all of the 8 bin histograms and create an $8 \times N_{bp}$ bin histogram for each frame. Then we apply two representations and show how our approach outperforms the state-of-the-art. As the first representation, we simply accumulate all the histograms assigned to each clip in one histogram. For the second representation, we want to model the temporal variation within the video. We employ the Fisher Kernel to do so.

The Fisher Kernel representation was introduced recently to improve the BoW for representing sets of local appearance descriptors. The Fisher Kernel was designed to combine the benefits of both *generative* and *discriminative* approaches [35] and creates a fixed-length representation for a set of vectors. In this research we use the Fisher Kernel to model the temporal variation in video. To do this, one can view a set of frame-based features (where we extract one feature from each frame) as a cloud of feature vectors. We can model this cloud with respect to a Gaussian Mixture Model (GMM) with diagonal covariance matrices. The resulting Fisher representation models the temporal variation in a generative way. Afterwards, we use the Fisher vector in a discriminative classifier (SVM).

The gradient vector is, by definition, the concatenation of the partial derivatives with respect to the model parameters. Let μ_i and σ_i be the mean and the standard deviation of i's Gaussian centroid, $\Gamma(i)$ be the soft assignment of descriptor x_t to Gaussian i, and let D denote the dimensionality of the descriptors x_t. $G^x_{\mu,i}$ is the D-dimensional gradient with respect to the mean μ_i and standard deviation σ_i of Gaussian i. Mathematical derivations lead to [70]:

$$G^x_{\mu,i} = \frac{1}{T\sqrt{\omega_i}} \sum_{t=1}^{T} \Gamma(i) \frac{x_t - \mu_i}{\sigma_i} \tag{4.7}$$

$$G^x_{\sigma,i} = \frac{1}{T\sqrt{2\omega_i}} \sum_{t=1}^{T} \Gamma(i) \left[\frac{(x_t - \mu_i)^2}{{\sigma_i}^2} - 1 \right] \tag{4.8}$$

where the division between vectors is a term-by-term operation. The final gradient vector G^x is the concatenation of the $G^x_{\mu,i}$ and $G^x_{\sigma,i}$ vectors, for $i = 1...K$. The final feature vector becomes a $2KD$ dimensional vector. At the end, we perform the normalization of the Fisher vectors since [70] has found this to significantly increase performance. The applied normalization is a combination of $L2$ and power normalization ($f(x) = sign(x)\sqrt{\alpha|x|}$) [70].

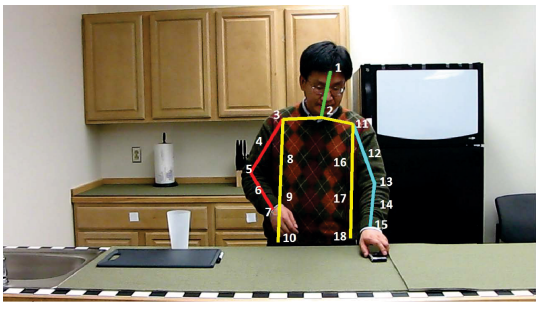
4.1.3 Results

Dataset

We present our pose estimation and activity recognition results on the ADL dataset. This dataset consists of 10 different activities: *answering a phone*, *dialing a phone*, *looking up numbers in a phone book, writing on a white board, drinking water, eating a snack, peeling a banana, eating a banana, chopping a banana* and *eating food with silverware*. Each of these activities is performed 3 times by 5 different people. These people have different genders, ethnicity, and appearance so sufficient appearance variation is available in the dataset. The original frame size is 1280×720. The frame-rate of the videos is 30 frames/s. Each clip is in the range of 3-50s and we extract features at a rate of one frame/s.

Groundtruth and Performance Evaluation

In this work, we provide qualitative analysis of our approach for body pose estimation and for activity recognition and we compare our results with related works. The ground truth for activity recognition comes with the dataset (*i.e.* each video clip contains a specific activity), but no groundtruth on the body pose is provided. Thus, we annotated 371 frames from different clips of ADL. For the annotation, we followed the procedure indicated in [141]. The example of an annotated frame is shown in Fig.4.2. Each of the 18 points in Fig. 4.2(a)

(a) 

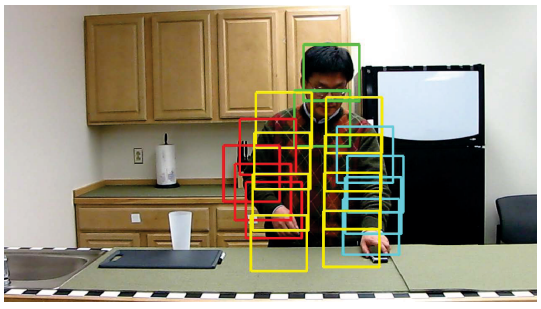
(b) 

Figure 4.2: A sample frame and its corresponding ground truth: (a) body pose tree showing the numbers in the correct positions (b) bounding boxes.

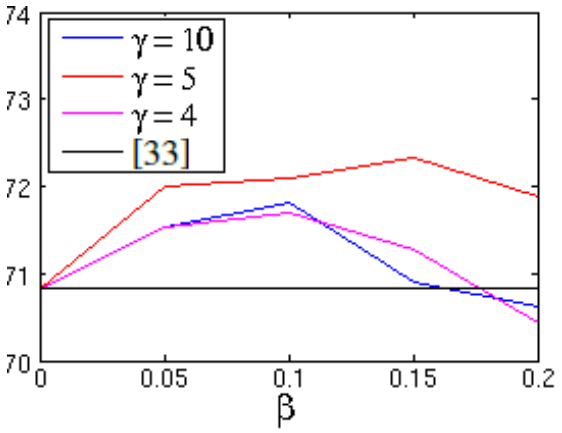
is the centroid of the bounding box of the corresponding body part of size L (as shown in Fig.4.2(b)).The accuracy of the body pose estimation is computed by comparing the positions of the groundtruth bounding box B_i^{GT} and of the estimated bounding box B_i^E, for each body part $i = 1, ..., 18$. If the overlap of B_i^E with B_i^{GT} is more than 80%, the body part is considered as being correctly detected. The accuracy of the body pose estimation is obtained by averaging over the accuracies of individual body parts.

Body Pose Estimation

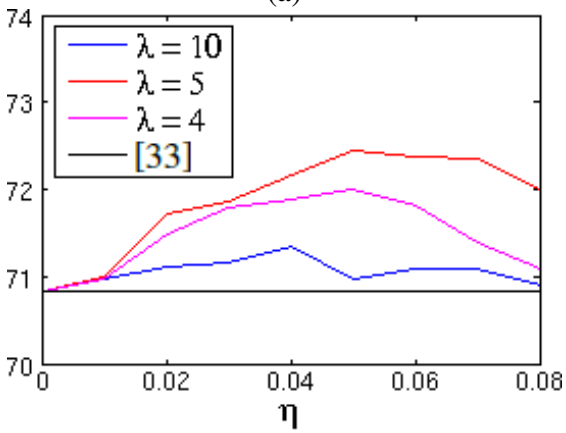
In Eq. (4.4), α is a parameter controling the relative importance of *foreground* and *optical flow* scores. To find the optimal values for different parameters, we tune parameters separately for S_{FG} and S_{OF}. To do so, we first set up $\alpha = 0$ and $\alpha = 1$ and find the optimum solution for (β, γ) and (η, λ), respectively. Then we tune α to get the best relative importance of S_{FG} and S_{OF}.

Varying parameters of S_{FG} and S_{OF}. Fig. 4.3(a) and Fig. 4.3(b) show how varying the parameters γ, β and λ, η changes the detector's performance. We recall that increasing γ and λ respectively decreases the widths of the foreground window and the optical flow window. Choosing a too small value for the parameters γ and λ consequently increases the size of the foreground and optical flow windows which worsen the detection results by bringing background noise into account. Choosing too large values for γ and λ decreases the size of the windows and consequently some low-level information related to the foreground and optical flows is discarded and hence the performance will decrease. In our experiment we found $\gamma = \lambda = 5$ as the best values, and consequently the foreground and optical flow windows have the same size.

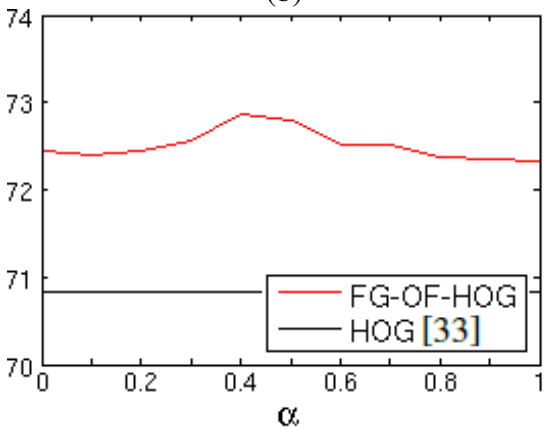
As we previously mentioned, β and η respectively represent the weights of the foreground and optical flow scores. Giving larger weights to the foreground or optical flow scores forces the detector to the ignore information that is ob-



(a)



(b)



(c)

Figure 4.3: Body parts detection accuracy at varying parameters (a) γ, β while $\alpha = 1$; (b) λ, η while $\alpha = 0$; (c) α.

Body part	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Average
HOG[141]	83.3	89.0	92.2	84.9	67.7	60.9	46.1	84.4	60.9	56.3	89.2	84.4	65.8	63.1	51.2	80.6	60.7	54.5	70.8
FG-HOG	83.7	88.4	93.3	84.6	67.9	58.2	43.7	**87.9**	**64.4**	**62.5**	90.0	**87.6**	67.1	**68.5**	55.5	80.6	60.9	**57.4**	72.3
OF-HOG	**83.8**	**89.0**	93.3	**85.2**	**68.7**	60.7	**48.5**	86.5	63.9	59.8	89.7	86.5	67.4	66.9	55.8	**81.4**	60.9	56.3	72.5
FG-OF-HOG	83.8	89.0	93.3	85.4	70.1	62.0	49.1	86.5	63.9	60.7	89.2	85.4	67.4	68.7	55.5	83.0	61.5	57.1	**72.9**

Table 4.1: Accuracy of different parts of the body. For most of the cases, applying FG-OF-HOG local descriptor achieves a better detection accuracy. The last column represents the overall performance. Bold numbers show which single-descriptor works better on the correspondent part.

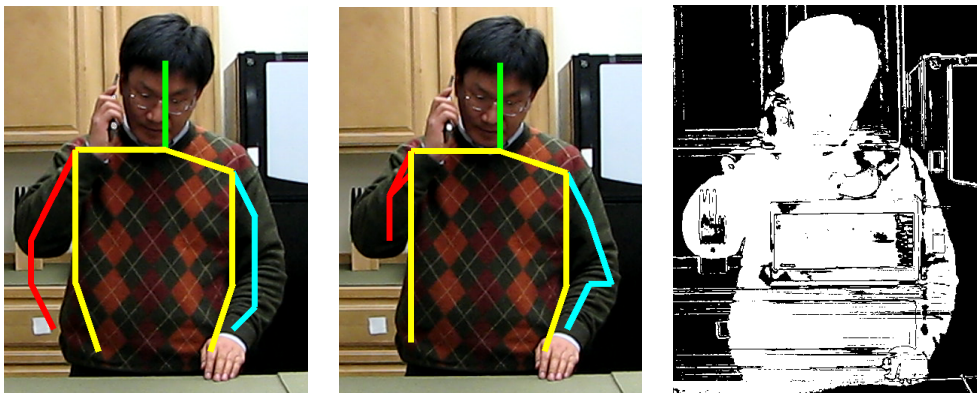
tained from HOG. In our experiments, we found that the best solution for these parameters is $\beta = 0.15$ and $\eta = 0.05$. In Fig. 4.3(c) we show how the relative pose estimation performance changes by giving different weights (α) to the foreground and optical flow scores. The highest performance is obtained by giving the weight 0.6 to the optical flow score and 0.4 to the foreground score (*i.e.* $\alpha = 0.4$).

Detection performance on different body-parts. Table 4.1 presents the best detection performance for different body parts using different local descriptors. Bolded numbers in Table 4.1 illustrate that applying the foreground descriptor improves the detection performance of the parts that are located in the subject's torso, while the optical flow score improves the detection performance mostly on the subject's hands as in the ADL dataset, usually the hands are moving more than the other parts.

Fig. 4.4(a) illustrates a sample in which using foreground information (Fig.4.4(c)) helps the detection of the right hand of the subject (Fig. 4.4(b)). Fig. 4.5(d) shows an example in which optical flow information (Fig. 4.5(c)) helps the body-pose estimator to detect the left hand of the subject in Fig. 4.5(b).

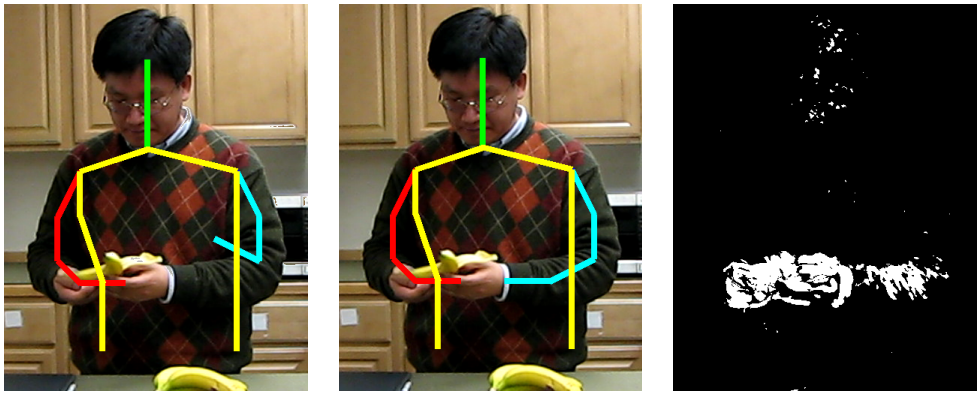
Activity recognition

In Table 4.2(a) we present the performance of our activity recognition approach for the 2 different representations (we use leave-one-person-out cross-validation):



(a) (b) (c)

Figure 4.4: Body configuration obtained with (a) [141] and (b) our method, including the information of the foreground mask in the body pose estimation (c).



(a) (b) (c)

Figure 4.5: Body configuration obtained with (a) [141] and (b) our method, including the information of the optical flow in the body pose estimation (c).

Local descriptor in body part detector	Accumulation	Fisher-Kernel
HOG [141]	87.32	95.71
FG-HOG	88.93	98.57
OF-HOG	87.50	97.14
FG-OF-HOG	89.11	98.75

(a)

Method	Accuracy
Wang *et al.* [132]	96.0
Bilen *et al.* [5]	74.0
Matikainen*et al.* [58]	70.0
Satkin *et al.* [94]	80.0
Bilinski*et al.* [6]	93.33
Kuehne*et al.* [43]	82.0
Messing *et al.* [60]	89.0
Our approach	**98.75**

(b)

Table 4.2: Activity recognition performance: (a) our approach: descriptor accumulation over a video sequence vs. Fisher Kernel representation for different body pose estimation methods (b) Performance comparison with the state-of-the-art on the ADL dataset.

(1) accumulate features descriptors over an entire video sequence and (2) use the Fisher-Kernel representation. The results show that even with the first representation that discards all the information about the temporal order and variation we obtain similar performance to some works in the literature that applied more expensive feature descriptors (see Table 4.2(b)). Additionally, by applying the Fisher Kernel representation we outperform all the state-of-the-art methods (Table 4.2(b)). The closest accuracy performance is reported by Wang *et al.* [132]. They applied Multi-Kernel-Learning, while our result is obtained using SVM with RBF kernel. The results with the Fisher Kernel representation are obtained with an optimized number of GMM centroids (the dictionary size), which in this case is equal to 20.

4.1.4 Conclusions and Future Work

In this work we present an efficient method to recognize activities of daily living. We combine the cues obtained from a body pose detector and local motion. This step created a descriptor that uses the structure of located motion. In this way, we involve high-level information combined with the low-level cues. Moreover, we show that including low-level cues (*i.e.* optical flow and foreground) together with an *off-the-shelf* body part detector gives a better performance without the need to re-train the detectors. In fact, we generate optical flow information once, and then apply it for both *enriching the body-part detector* and *quantizing flows* for activity recognition task. We also model the temporal variation within the video using the Fisher Kernel representation. Finally, our novel descriptor with the Fisher Kernel representation achieved the best reporting results so far for the ADL dataset. In future work we plan to extend our approach for more challenging scenarios such as *fine-grained activities* [82].

4.2 It's all about Habits: Exploiting Multi-Task Clustering for Activities of Daily Living Analysis

Activities of daily living (ADL) are defined as "the things we normally do on a daily basis for self-care such as feeding ourselves, bathing, dressing, grooming, work, homemaking, and leisure"[4].
In the last few years, automatic analysis of ADL has received an increasing interest in the computer vision and image processing community [60, 82], mainly due to need of developing innovative tools for applications such as patient monitoring, tele-rehabilitation and, more in general, ambient assisted living.
Several works have been proposed to face the multiple challenges of recognizing complex activities of everyday life in different scenarios (*e.g.* kitchen, office, home) [60, 63, 82, 112]. More recently monitoring of activities of daily living has gained importance also in the context of the so called "life-logging" applications, *i.e.* when a first-person camera continuously records a whole day of its wearer life [54, 72]. The problem of everyday activity recognition poses several challenges, mostly implying the engineering of discriminative features and scalable recognition algorithms.
A further problem arises as, independently of the considered scenario, several hours of videos are usually collected. This generates a large amount of data for which annotation is typically not available requiring a great human labeling effort.

In this study, we consider the problem of everyday activity recognition from unlabeled data under a novel perspective. Our intuition is that ADL analysis is intrinsically a multi-task learning problem. People working in an office environment tend to perform the same kind of activities (*e.g.* working in front of a personal computer, reading documents). Similarly, most of people when they

[4] http://en.wikipedia.org/wiki/Activities of daily living#cite note-MN-2

wake up in the morning use to drink coffee and brush their teeth. Fig 4.2 depicts an overview of the considered problem.

Thus, it is intuitive that, when performing activity recognition, learning from data of all the subjects simultaneously is advantageous with respect to learning from data of each person separately. In other words, it is evident that the tasks of recognizing activities of each single subject are related. However, the data distributions can be different, since visual data corresponding to different people may exhibit different features. In particular if there are limited data for a single person, typical clustering methods may fail to discover the correct clusters. In this case, using data from other subjects as auxiliary source of informations will induce the correct clusters. However, simply combining data from different people together and applying traditional clustering approach does not necessarily lead to performance improvement, because the data distributions of single tasks can be different, violating *i.i.d.* assumptions. To address this problem, we propose to invoke the novel paradigm of multi-task clustering.

Contributions: To summarize, the contributions of this work are the following:

- We address the problem of everyday activity recognition by proposing a multi-task clustering approach.
- Two novel methods are introduced, derived by a common framework based on the minimization of an objective function balancing two terms, one which ensure the data of each single task to be clustered appropriately, the other which enforce some coherence between the clustering results of related tasks.
- For both the proposed multi-task clustering algorithms an efficient solver is derived.
- We demonstrate the effectiveness of our approaches on Rochester Activity

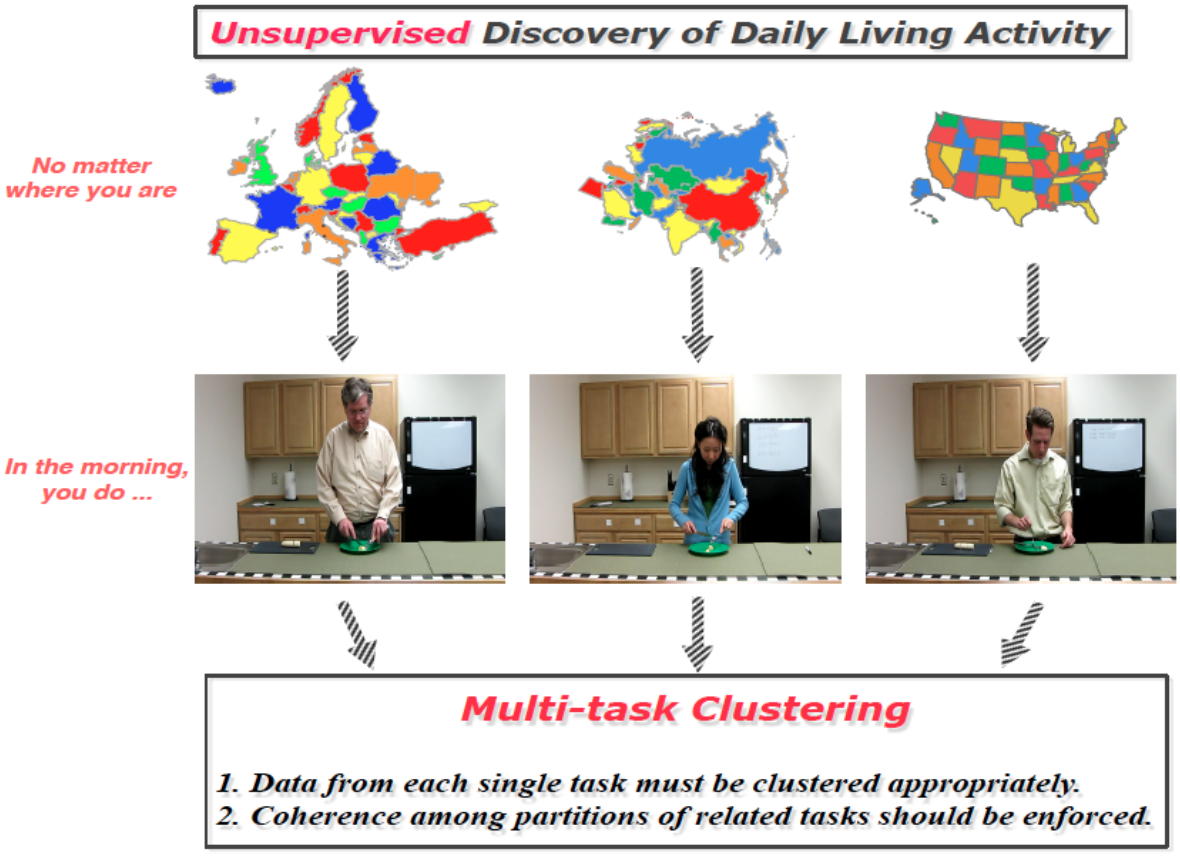


Figure 4.6: Overview of the considered problem: no matter where you are, in the morning you probably have breakfast and use a knife to cut food to pieces. In this work we exploit this and other informations about people habits to perform ADL analysis proposing a novel multi-task clustering approach.

of Daily Living dataset, comparing them with several single task and multi-task learning methods.

4.2.1 Related Work

We review prior works in the context of activity of daily living analysis and multi-task clustering.

Activity of Daily Living Analysis. In the last few years several works have considered the problem of everyday activity recognition, not only in com-

puter vision but also in other related research areas, *e.g.* ubiquitous computing [47, 113].

Many of these recent works are based on the use of RFID tags or inertial sensors. However, systems based on cameras still have an important role since they are generally cheap and easy to deploy. A survey of recent works on activity recognition in computer vision is presented in [117]. Some recent publications have addressed specifically the task of ADL analysis [60, 72, 47]. Messing *et al*. [60] proposed an approach based on features considering velocity history of tracked keypoints for recognizing complex everyday activities performed in a kitchen environment. In [82], Rohrbach *et al*. also considered a kitchen scenario but focused on the most difficult problem of fine-grained activity recognition. Other works (see *e.g*. [47]) exploited the use of the novel RGB-D sensors showing improved performance with respect to traditional cameras alone. In [72], the authors considered the more challenging task of everyday at home activity recognition from a wearable camera and demonstrated the importance of using features based on object detectorsoutput in such uncontrolled scenario. In this work we address the problem of analysing activities of daily living under the perspective of multi-task learning. However, while multi-task learning have already been exploited in the context of visual based activity recognition [55, 139, 144], we are not aware of works which address simultaneously the challenge of lack of annotated data. This is particularly important especially in the perspective of analyzing streams in the context of life-logging, when annotated data are difficult to get.

Multi-task Learning. Multi-task learning approaches have received considerable attention in the last few years [12]. Learning from data of multiple related tasks simultaneously is greatly advantageous in terms of performance with respect to learning on every single task independently. The effectiveness of multi-task learning have been demonstrated also in several applications in computer vision, such as object detection [90], object classification [32] or face verifica-

tion [134]. Most existing works on multi-task learning methods tackle classification and regression problems. Only few works have considered unsupervised approaches to multi-task learning [31, 44, 147], *i.e* the scenario where the data of each task are unlabeled and the aim is to predict the cluster labels in each task. In [31], the authors proposed to learn a subspace shared by all the tasks, through which the knowledge of one task can be transferred to all the others. Zhang and Zhang [147], introduced a multi-task clustering approach based on a pairwise agreement term which encourage coherence among clustering results of multiple tasks. In [44] the *k*-means algorithm is revised from a Bayesian nonparametric viewpoint and extended to multi-task learning.
In this study, we follow these recent works and propose two novel approaches for multi-task clustering. The first one is inspired by the work in [147], but it is based on another objective function and thus on a radically different optimization algorithm. Furthermore in the considered application it guarantees superior accuracy with respect to the method in [147]. Our second approach results into a convex optimization problem, thus avoiding the issues related to local minima of all the methods derived by *k*-means relaxations.

4.2.2 Multi-task Clustering

We are given T related data sources, each one consisting of data samples in the set $X_t = \left\{x_1^t, x_2^t, ..., x_{N_t}^t\right\}$, where $x_j^t \in \mathbb{R}^d$ is a d-dimensional feature vector, N_t is the number of samples associated to the t-th data source (task). We want each data source to be partitioned into k_t clusters, where the number of required partitions can be different in different tasks. As we assume the tasks to be related, we also require that the resulting partitions are consistent with each other. This can be obtained by defining the following optimization problem:

$$\min_{\Theta_t} \sum_{t=1}^{T} \wedge (X_t, \Theta_t) + \sum_{t=1}^{T} \sum_{s=1}^{T} R(\Theta_t, \Theta_s) \tag{4.9}$$

Eq. 4.9 corresponds to the general problem of Multi Task Clustering (MTC), where the term $\wedge(\cdot)$ corresponds to a reconstruction error which must be minimized by learning the optimal model parameters Θ_t (*i.e.* typically consisting in the cluster centroids and the associated assignment matrix), while $R(\cdot)$ is an agreement term, imposing that, since the multiple tasks are related, also the associated model parameters should be similar. Under this framework, in this work we propose two different approaches for MTC. The first one extends the method proposed in [147], as we also adopt the Earth Mover's Distance function to define the models'agreement term $R(\cdot)$. However since a different function is used for $\wedge(\cdot)$, a different algorithm with respect to the one described in [147] is proposed for solving Eq. 4.9. Furthermore, our experimental results demonstrate improved performances with respect to [147]. A second approach we introduce is based on a different function $R(\cdot)$ to measure tasks models consistency and it ultimately results into a convex optimization problem which can be solved efficiently with the alternating direction method of multipliers (ADMM) [9]. In the following subsections we introduce the proposed methods.

EMD Regularized Multi-task Clustering

We consider the data matrix $X \in \mathbb{R}^{(N_1,N_2,\cdots,N_T)\times d}$, $X = [X_1 X_2 \cdots X_T]$ obtained concatenating the individual matrices $X_t = \left[x_1^t x_2^t \cdots x_{N_t}^t\right]$ associated to each task t. We are interested in finding the centroid matrix $C_t = [C_1 C_2 \cdots C_t]$, $C \in \mathbb{R}^{(k_1+k_2+\cdots+k_T)\times d}$, $C_t \in \mathbb{R}^{k_t \times d}$, , and the cluster indicators block diagonal matrix $W = blkdiag(W_1, W_2, \cdots, W_T)$, $W \in \mathbb{R}^{(N_1+\cdots+N_T)\times(k_1+\cdots+k_T)}$, $W_t \in \mathbb{R}_t^N \times k_t$, by solving the following optimization problem:

$$\min_{\substack{C_1,\cdots,C_T \\ W_1,\cdots,W_T, f_{ij}\geq 0}} \sum_{t=1}^{T} \|X_t - W_t C_t\|_F^2 + \lambda \sum_{t,s=1}^{T} \sum_{i=1}^{k_t} \sum_{j=1}^{k_s} f_{ij} [(C_t)i. - (C_s)j.]' [(C_t)i. - (C_s)j.]$$

$$s.t. \begin{cases} \sum_{j=1}^{k_s} f_{ij} = \sum_{n=1}^{N_t} (W_t)_{ni} & (1 \leq i \leq k_t) \\ \sum_{i=1}^{k_t} f_{ij} = \sum_{n=1}^{N_s} (W_s)_{nj} & (1 \leq j \leq k_s) \\ \sum_{i=1}^{k_t} \sum_{j=1}^{k_s} f_{ij} = 1 & (1 \leq i \leq k_t, 1 \leq j \leq k_s) \end{cases} \tag{4.10}$$

where $(\cdot)'$ denotes the transpose operator, $(C_t)_{i\cdot}$ and $(C_t)_{j\cdot}$ denote the i-th row of C_t and j-th row of C_s respectively. The first term in the objective function is a relaxation of the traditional k-means objective function for T separated data sources. The second term is added to explore the relationships between clusters of two different data sources and it consists of the popular Earth Movers Distance [87] computed considering the signatures ς and τ obtained by clustering the data associated to task t and s separately, *i.e.* $\tau = \left\{((C_t)_{1\cdot}, w_t^1), \cdots, ((C_t)_{k_t\cdot}, w_t^{k_t}))\right\}$, $w_t^i = \sum_{n=1}^{N_t} (W_t)_{ni}$ and $\varsigma = \left\{((C_s)_{1\cdot}, w_s^1), \cdots, ((C_s)_{k_s\cdot}, w_s^{k_s}))\right\}$, $w_s^i = \sum_{n=1}^{N_s} (W_s)_{ni}$. In practice $(C_t)_{i\cdot}$ and $(C_s)_{j\cdot}$ are the cluster centroids and w_i^s, w_i^t denote the weights associated to each cluster (reflecting somehow the number of datapoints associated to each cluster). In practice the second term represents a distance between two distributions and minimizing it we impose the found partitions between two related tasks to be consistent.

In (4.10) there are no constraints on the C values. In this work, we also impose that the vectors definiting C lie within the column space of X, *i.e.* the columns of C are a weighted sum of certain data points. In other words, we define $C = PX$ where $P = \text{blkdiag}(P_1 \cdots P_T)$, $P \in \mathbb{R}^(k_1 + \cdots + k_T) \times (N_1 + \cdots + N_T)$. In the following, for sake of simplicity and easy interpretation, we consider only a two tasks problem. The extension to T tasks is straightforward. The optimization problem we consider is:

$$\begin{aligned} \min_{P>0, W>0, F>0} \quad & \|X - WPX\|_F^2 + \lambda tr(MPXX'P'M'F) \\ \text{s.t.} \quad & \|(P_t)_{\cdot i}\| = 1, \forall i \quad \forall t \in \{1, 2\} \end{aligned} \tag{4.11}$$

$$
\begin{aligned}
tr(I_j F) &= \sum_{i=1}^{N_1+N_2} W_{ij}, j = 1, \cdots, k_1 + k_2 \\
tr(F) &= 1
\end{aligned}
\tag{4.12}
$$

where F =diag$(f_{11}, \cdots, f_{k_1 k_2})$, $F \in \mathbb{R}^{k_1 k_2 \times k_1 k_2}$, and $I_j \in \mathbb{R}^{k_1 k_2 \times k_1 k_2}$ and $M \in \mathbb{R}^{k_1 k_2 \times (k_1 k_2)}$ are appropriately defined selection matrices $I_j = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$,

$$
M = \begin{bmatrix} 1 & 0 & 0 & \dots & -1 & 0 & \dots \\ 1 & 0 & 0 & \dots & 0 & -1 & \dots \\ 1 & 0 & 0 & \dots & 0 & \dots & -1 \\ 0 & 1 & 0 & \dots & -1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \\ 0 & 0 & 1 & \dots & 0 & \dots & -1 \end{bmatrix}.
$$

Optimization

To solve the proposed problem (4.11), we first note that the optimal solution of (4.11) can be found adopting an alternating optimization scheme, *i.e.* optimizing separately (4.11) first with respect to P and then with respect to W and F jointly. In both cases, a non-negative least square problem with constraints arises, for which standard solvers can be employed. However, due to computational efficiency, here we consider an approximation of (4.11), replacing the constraints (4.12) with $tr(I_j F) = e$, where $e \in \mathbb{R}^{k_1 k_2}$, $(e)_i = \frac{1}{k_1}$, if $i \leq k_1$, $(e)_i = \frac{1}{k_2}$ otherwise. This approximation in practice implies that for each task the same number of datapoints is assigned to all the clusters. Furthermore a more efficient solver can be devised. Specifically, we adopt an alternating optimization strategy, *i.e*, we optimize separately 4.11 with respect to P, W and F until convergence as explained in the following:

Step 1: Fixed W, P, optimize F.

$$\begin{aligned} \min_{F>0,\ \ tr(F)=1} \quad & \lambda tr(MPXX'P'M'F) \\ \text{s.t.} \quad & tr(I_j F) = e, j = 1, \cdots, k_1 + k_2 \end{aligned} \tag{4.13}$$

This is a simple Linear Programming (LP) problem than can be solved effectively with standard solvers.

Step 2: Fixed F, P, optimize W,

$$\min_{W>0} \|X - WPX\|_F^2$$

Following [50], we update W using a projected gradient method for bound-constrained optimization, *i.e.* using $W^{k+1} = \max(0, W^k - \alpha_k \nabla_W \wedge (P^k, W^k, F^k))$, where $\nabla_W \wedge (P^k, W^k, F^k) = WPXX'P' - XX'P'$.

Step 3: Fixed W, F, optimize P.

$$\min_{P>0} \|X - WPX\|_F^2 + \lambda tr(MPXX'P'M'F)$$

$$\text{s.t.} \left\|(P_t)_{.i}\right\|_1, \forall i \ \ \forall t = 1, 2$$

Similarly to step 2 we update P using a projected gradient method for bound-constrained optimization *i.e.* $P^{k+1} = \max(0, P^k - \alpha_k \nabla_P \wedge (P^k, W^k, F^k))$, where $\nabla_P \wedge (P^k, W^k, F^k) = W'WPXX' - W'XX' + \lambda M'FMPXX'$. To account for constraints at each itera- tion we also normalize each column of P, following the nor- malization invariance approach of Eggert and Korner [23].

Kernalization

Finally, to kernelize the proposed method, we consider a mapping $X \to \phi(X)$. The objective function of problem (4.11) becomes:

$$\begin{aligned} &\|\phi(X) - WP\phi(X)\|_F^2 + \lambda tr(MP\phi(X)\phi'(X)P'M'F) \\ &\quad = Tr(\phi(X)\phi(X)' - 2\phi(X)\phi(X)'P'W' \\ &+WP\phi(X)\phi(X)'P'W' + \lambda MP\phi(X)\phi(X)'P'M'F) \end{aligned}$$

It is evident that the latter only depends on the kernel matrix $K = \phi(X)\phi(X)'$. The update rules of

the kernalized version of our method can be easily derived from the previous subsection, using $\phi(X)\phi(X)'$ instead of $X'X$.

Convex Multi-task Clustering

Given the data samples $X_t, t = 1, \cdots, T$ we propose to learn the sets of cluster centroids $\Pi_t = \left\{\pi_1^t, \pi_2^t, \cdots, \pi_{N_t}^t\right\}$, $\Pi_t \in \mathbb{R}^{N_t \times d}$ associated to each task t by solving the following optimization problem:

$$\begin{aligned} \min_{\pi_i^t} \sum_{t=1}^{T} \Big(\sum_{i=1}^{N_t} \|x_i^t - \pi_i^t\|_2^2 + \lambda_1 \sum_{i=1}^{N_t} \sum_{j=i+1}^{N_t} w_{ij}^t ||\pi_i^t - \pi_j^t||_1 \Big) \\ + \lambda_2 \sum_{t,s=1}^{T} \gamma_{st} \sum_{i=1}^{N_t} \sum_{j=1}^{N_s} ||\pi_i^t - \pi_j^s||_2^2 \end{aligned} \tag{4.14}$$

In 4.14, the first two terms guarantees that the data of each single task are clustered, while the last term imposes the found centroids to be similar if the tasks are related. The relatedness between tasks is modeled by the parameter γ_{st}, which can be set using a measure between distributions (we consider the Maximum Mean Discrepancy proposed in [8]) reflecting the relatedness among tasks. The parameter w_{ij}^t is used to enforce datapoints of the same task to be assigned to the same cluster. In this project, we set $w_t^{ij} = \tau_t e^{-\lambda ||x_i^t - x_j^t||^2}$ if $e^{-\lambda ||x_i^t - x_j^t||^2} \leq \theta$ and $w_{ij}^t = 0$ otherwise. The paramethers λ_1 and τ_t are used to control the number of clusters, while the term $e^{-\lambda ||x_i^t - x_j^t||^2}$ is considered as in this way the found partition structures takes into account the density of the original data distributions.

Optimization

To solve 4.14 we propose a method based on alternating direction method of mutlipliers (ADMM) [9]. We consider the data matrix $X = [X_1 \cdots X_T]$ and the centroid matrix $\Pi = [\Pi_1 \cdots \Pi_T]$ obtained concatenating matrix specific tasks. We notice that the problem is separable on d dimensions, *i.e.* solving (4.14) just

amounts to solving d separeate minimization subproblems as follows:

$$min_{\Pi^d} \quad ||X^d - Pi^d||_2^2 + \lambda_1 ||E\Pi_d||_1 + \lambda_2 ||B\Pi^d||_2^2 \tag{4.15}$$

where X^d and Π^d represent the vectors corresponding to the d-th row of the matrices X and Π, the matrix E is a block diagonal matrix defined as $E = (blkdiag)(E_1, E_2, \cdots, E_T)$ and $E_t \in \mathbb{R}^{|\varepsilon| \times N_t}$ is a matrix with $|\varepsilon_t| = \frac{N_t(N_t-1)}{2}$ rows and each row is a vector of all zeros except in the position i, where it assumes the value w_{ij} and in the position j where it has the value $-w_{ij}$. Similarly the matrix $B \in \mathbb{R}^{|B| \times N}$, where $|B| = \frac{T(T-1)}{2}$ and N is the number of datapoints for all the available tasks, have the aim to impose a smoothness terms between tasks. A row of the matrix B is a vector with all zeros except to the terms corresponding to datapoints of the t task which are set to λ_{st} and to the terms corresponding to datapoints of the $s-$th task which are all set to $-\gamma_{st}$.

To solve (4.15) we consider the equivalent constrained optimization problem:

$$\begin{aligned} &\min_{\Pi^d} ||X^d - \Pi^d||_2^2 + \lambda_2 ||B\Pi^d||_2^2 \\ &\text{s.t.} \quad E\Pi^d - q = 0 \end{aligned} \tag{4.16}$$

The associated lagrangian is:

$$\begin{aligned} L_p(\Pi^d, q, p) = ||X^d - \Pi^d||_2^2 + \lambda_2 ||B\Pi^d||_2^2 \\ + P'(E\Pi^d - q) + \tfrac{\rho}{2} ||E\Pi^d - q||_2^2 \end{aligned}$$

with P being the vector of augmented Lagrangian multipliers and ρ being the dual update step length. In the ADMM, three steps, corresponding to the update of three variables, are performed:

Step1: Update Π^d, given q, p fixed, by solving:

$$\begin{aligned} [\Pi^d]^{k+1} = \arg\min_{\Pi^d} \quad & ||X^d - \Pi^d||_2^2 + \lambda_2 ||B\Pi^d||_2^2 \\ & + (E'p^k)'\Pi^d + \tfrac{\rho}{2} ||E\Pi^d - q||_2^2 \end{aligned} \tag{4.17}$$

Imposing the gradient with respect to Π^d equal to 0, we get update step as:

$$M[\Pi^d]^{k+1} = b^k$$

where $M = \rho E'E + 2I + 2\lambda_2 B$ and $b^k = \rho E'q^k + 2X^d$. The computation of Pi^d involves solving a linear system. To solve it efficiently, we use Cholesky factorization decomposing $M = \wedge'\wedge$. In practice, at each FISTA iteration, we solve two linear systems: $\wedge' g = b^k$ and $\wedge\Pi^d = g$. Since $\wedge$ is an upper triangular matrix, solving these two linewar systems is efficient.

Step 2: Update q, given Pi^d, p fixed, by solving:

$$q^{k+1} = \arg\min_q \lambda_1 ||q||_1 - (p^k)'q + \frac{\rho}{2}||E[\Pi^d]^{k+1} - q||_2^2$$

Neglecting the constant terms, the update step is:

$$q^{k+1} = \arg\min_q \frac{1}{2}||q - E[\Pi^d]^{k+1} - \frac{1}{\rho}p^k||_2^2 + \frac{\lambda_1}{\rho}||q||_1$$

This equation has a closed-form solution. Defining the soft-thresholding operator $ST_\lambda(x) = \text{sign}(x)\text{max}(|x| - \lambda, 0)$ we get:

$$q^{k+1} = ST_{\lambda_1/\rho}(E[\Pi^d]^{k+1} + \frac{1}{\rho}p^k)$$

Step 3: Update p, given Π^d, q fixed, with the equation:

$$p^{k+1} = p^k + \rho(E[Pi^d]^{k+1} - q^{k+1})$$

4.2.3 Activity of Daily Living Analysis

The growing interest in the vision community towards novel approaches for Activities of Daily Living analysis has led to the realization of several datasets [60, 72, 82] publicly available for research purposes. To demonstrate the effectiveness of our Mutli-Task Clustering approaches on everyday activity recognition we specifically consider the ADL dataset from Rochester University [60]. This

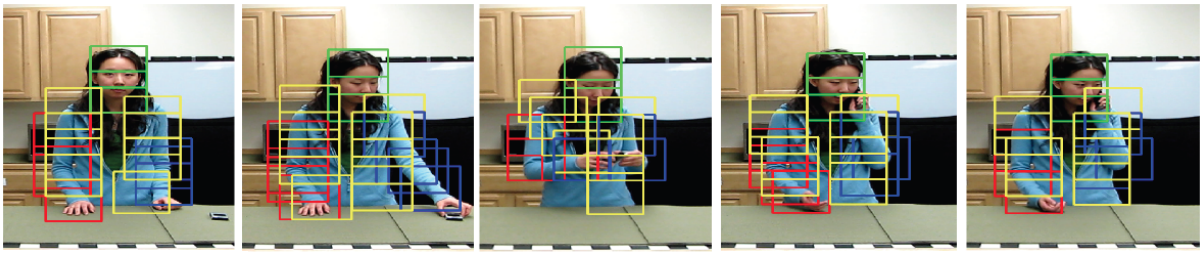


Figure 4.7: Rochester ADL dataset: a sequence depicting the activity answering phone and the computed body parts.

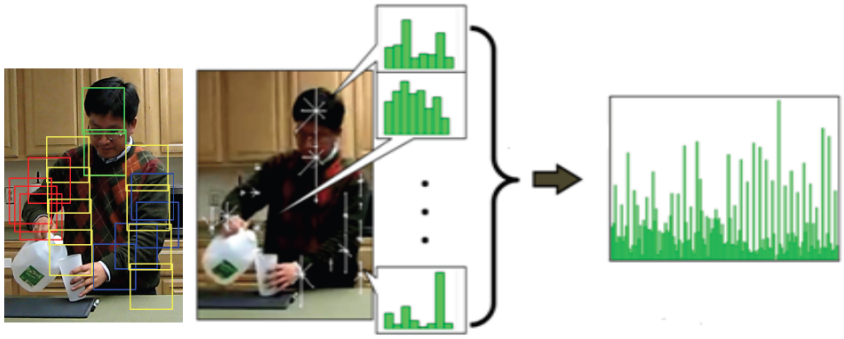


Figure 4.8: Rochester ADL dataset: feature representation for a single frame.

dataset consists of a set of pre-segmented video clips, each depicting 10 different activities performed 3 times by 5 different subjects. Typical activities are answering a phone, drinking water, eating a snack, or peeling a banana. The considered people have different appearance, genders and ethnicity. Each video clip is on average 3-50s long. The frame size is 1280×720 and the frame rate is 30 frames per second.

Feature Extraction

We follow one of the most recent works on this dataset [60] and we first extract features on a frame-basis (at rate of one frame/s) considering a combination of both low-level and high-level cues. Specifically to compute high-level cues we adopt the pictorial deformable model for body pose estimation proposed in [63] and detect the location of 18 body-parts. Fig. 4.8 shows an example of body

parts extracted on a sequence of the activity answering phone. To extract low-level cues, we compute the optical flow using the Lucas-Kanade algorithm and we quantize it into 8 possible directions. Then we construct a descriptor for each body part, represented by an eight bin histogram computed from the optical flow information. Finally, we concatenate all the histograms and create a 144 bin histogram for each frame (Fig. 4.2.3). To compute the video clips descriptors we adopt two different approaches as suggested in [60] one consisting in accumulating frame features, the other in using a Fisher-Kernel representation.

Experimental results

We perform a series of experiments randomly selecting two targets out of five from the Rochester ADL dataset. Thus, two tasks are considered in our MTC approaches. Experiments are run 10 times and the average results are reported. Table 4.3 and 4.4 show the results of different clustering methods applied on the the accumulation and the fisher kernel representations respectively.
We compare our approaches (EMD Regularized Multi-task Clustering with linear and rbf kernel and Convex Multi-task Clustering denoted as CEMD-MTC, KCEMD-MTC, CMTL respectively) with single task clustering approaches, *e.g.* the k-means (KM), kernel *k*-means (KKM), convex (CNMF) and semi (SemiNMF) nonnegative matrix factorization [21]. We also consider recent multi-task clustering approaches such as the SemiEMD-MTC proposed in [147], its kernel version KSemiEMD-MTC and the LSMTC method in [31]. For all the methods except that for our convex approach CMTL ten runs are performed corresponding to different initializations conditions. For each experiment the average results are considered. To evaluate the clustering results, we adopt the popular clustering accuracy (Acc) and normalized mutual information (NMI) metrics [137]. In CMTL the parameters λ_1 is set to 1 while the task specific τ_t are varied in order to obtain the desired number of clusters. The value of the regularization parameters of our approaches (λ for the approaches based on EMD

regularization and λ_2 for CMTL) are set in the range $\{10^{-2}, 10^{-1}, \cdots, 10^{2}\}$. The results reported in Table 4.9 and 4.10 correspond to the best clustering performance. Observing Table 1 and 2 several observations can be made. First of all, independently on the adopted features representation, multi-task clustering approaches always perform better than single task clustering methods (*e.g.* SemiEMD-MTC outperforms SemiNMF, CEMD-MTC provide higher accuracy than CNMF, a value of λ_2 greater than 0 lead to an improvement in accuracy and NMI in CMTC). An exception in this sense is represented by the LSMTC proposed in [31], which performs quite poorly (worse than *k*-means) in the considered application.

Confirming the findings reported in [72], we also observe that the Fisher Kernel representations is advantageous with respect to features computed based on a simple accumulation scheme. Noticeably, our methods are among the best performers, with KCEMD-MTC reaching the higher values of accuracy and NMI. This is somehow expected probably due to both the use of kernels and the adoption of the multi-task paradigm.

Finally, we also investigate the effect of different values of the regularization parameter λ in (3) on clustering performance when Fisher Kernel features are used. As shown in Fig.5, both accuracy and NMI values are sensitive to varying λ. The best performance for CEMD-MTC and KCEMD-MTC are obtained when $\lambda = 1$ and $\lambda = 0.1$ respectively. This clearly confirms the advantage of using a MTC approach for ADL analysis.

4.2.4 Conclusions

In this research we consider the task of everyday activity recognition from unlabeled data as a MTC problem. A novel MTC algorithm has been proposed and evaluated extensively on Rochester ADL dataset. Our results clearly demonstrate the advantage of using a MTC approach (in particular KCEMD- MTC) for ADL analysis. Future works include exploiting the suitability of the pro-

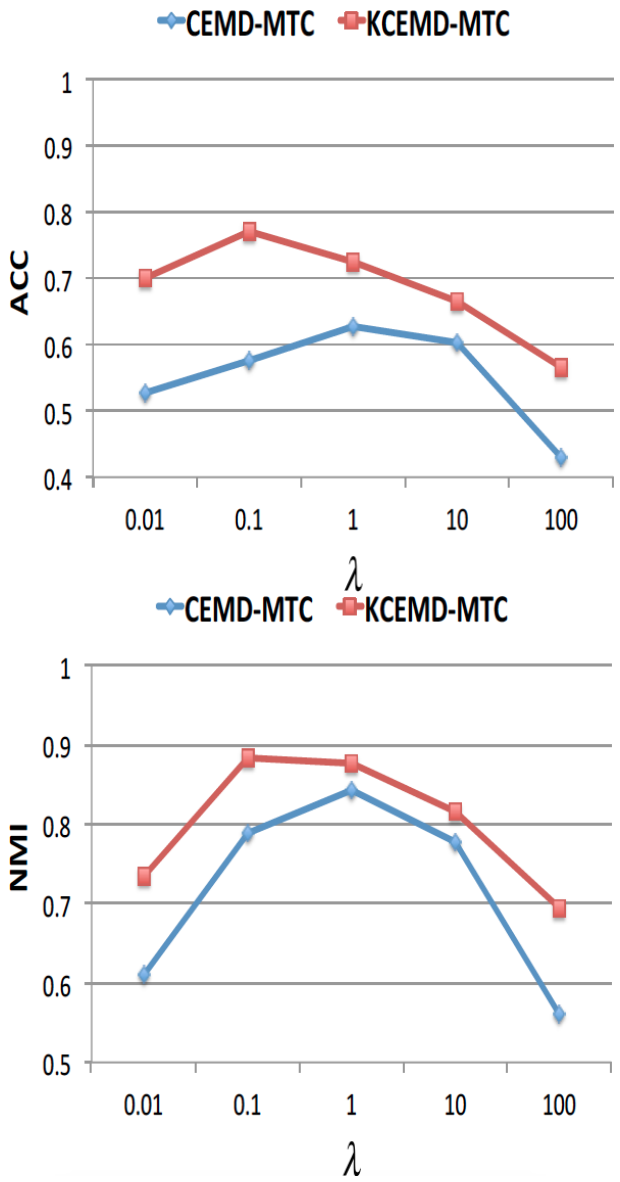


Figure 4.9: Performance variation at different value of for Task 2 of the Rochester ADL dataset.

	Acc			NMI		
	Task 1	Task 2	Avg	Task 1	Task 2	Avg
CMTC	0.567	0.602	0.585	0.682	0.673	0.678
CMTC ($\lambda_2 = 0$)	0.551	0.593	0.572	0.676	0.666	0.671
KM	0.523	0.513	0.518	0.671	0.646	0.659
KKM	0.545	0.537	0.541	0.689	0.672	0.681
SemiNMF[21]	0.556	0.526	0.541	0.604	0.637	0.621
SemiEMD-MTC[147]	0.580	0.533	0.557	0.658	0.655	0.657
KSemiEMD-MTC[147]	0.602	0.561	0.581	0.686	0.689	0.688
LSMTC[31]	0.480	0.503	0.492	0.598	0.621	0.610
CNMF[21]	0.607	0.647	0.627	0.746	0.772	0.759
CEMD-MTC	0.693	0.627	0.660	0.827	0.842	0.835
KCEMD-MTC	**0.700**	**0.770**	**0.735**	**0.853**	**0.883**	**0.868**

Table 4.3: Clustering results on Rochester ADL dataset: comparison of different methods using accumulation features.

	Acc			NMI		
	Task 1	Task 2	Avg	Task 1	Task 2	Avg
CMTC	0.604	0.612	0.608	0.691	0.685	0.688
CMTC ($\lambda_2 = 0$)	0.592	0.601	0.597	0.683	0.675	0.679
KM	0.533	0.537	0.535	0.682	0.656	0.669
KKM	0.555	0.552	0.554	0.704	0.694	0.699
SemiNMF[21]	0.581	0.531	0.556	0.634	0.639	0.637
SemiEMD-MTC[147]	0.595	0.567	0.581	0.678	0.675	0.677
KSemiEMD-MTC[147]	0.621	0.584	0.603	0.699	0.702	0.701
LSMTC[31]	0.501	0.525	0.513	0.602	0.634	0.618
CNMF[21]	0.621	0.644	0.633	0.755	0.782	0.769
CEMD-MTC	0.713	0.653	0.683	0.833	0.852	0.843
KCEMD-MTC	**0.741**	**0.765**	**0.753**	**0.874**	**0.888**	**0.881**

Table 4.4: Clustering results on Rochester ADL dataset: comparison of different methods using fisher kernel features.

posed MTC algorithms for other vision applications as well as investigating how to improve our MTC methods (*e.g.* by detecting outlier tasks).

Chapter 5

Conclusion and future work discussion

Rapid developments in computer vision and artificial intelligence at large are increasing the number of domains these techniques can be applied to. This dissertation[1] discussed the main challenges that are faced while performing one important computer vision task: video understanding. The 3 dissertation chapters outlined the most important questions in this field, proposed solutions, and potential applications.

The first part of Chapter 2 presented methods for functional traffic scene categorization by leveraging semantic information from high-level detectors to form video descriptors. This method was also used to perform fine-grained activity recognition in videos (explored in Chapter 4) and segment traffic scenes into semantic regions based on objects functions (in Chapter 2). In the fine-grained activity recognition task (for daily activities), the camera is static and one only needs to analyze the movements of objects interacting with the persons body parts. In addition, most activities are difficult to distinguish and those with different labels can be performed in a very similar manner. we presented a method that leverages semantic-based prior knowledge to distinguish between such similar actions. Using this high-level knowledge is particularly useful when there is limited labeled data (which is often the case in the fine-grained scenario).

[1]The author of this dissertation also published other papers on video understanding during her Ph.D. which are not presented in this dissertation [1, 86, 120]

Since human body parts are an important source of knowledge for identifying daily living activities, our method detects important limbs and builds descriptors based on their movements. Finally, we discuss important applications of daily living activity analysis ranging from patient monitoring to assisted living. Another source of semantic information that we address in this dissertation is how to adequately embed motion information for the task of human body pose estimation, which is discussed in the second part of the Chapter 2.

The above mentioned method only works well when one has strong prior knowledge regarding a particular scene. For example, in the fine-grained activity recognition scenario, the camera was expected to be static. In the traffic scene categorization task, the video is expected to contain pedestrians and vehicles. When the scene contains unexpected information and context, leveraging prior knowledge may not be very helpful. For instance, in Chapter 2, including knowledge pertaining to wrist movement resulted in significant improvement on the Rochester Pose ADL dataset while adding minimal improvement on the Pose in the Wild dataset. The latter contains many unexpected movements in each scene. Thus, in future work, we plan to model important semantic information without using prior knowledge about each scene.

One suggestion for incorporating semantic information is to use Attention Mechanisms. Attention has been shown to work well in many applications ranging from sequence to sequence tasks in natural language processing to image classification and captioning in computer vision. However, attention mechanisms have not garnered much success in tasks involving videos. Our work employed attention to perform classification on synthetic data generated for evaluating this task [2]. This synthetic data, called the cluttered bar mnist video dataset, mimics the features of real world video data. It is difficult, if not impossible to infer class labels from single frames–one has to watch the entire video to predict digit

[2]All code for employing attention and creating the dataset is available in my GitHub repo: https://github.com/negar-rostamzadeh/LSTM-Attention

labels. The attention-based model that we explored on synthetic data, was not only able to classify digits in this challenging scenario but also localized and tracked the objects of interest without any prior knowledge regarding the location or scale of the digits.

The richness of video data (containing temporal and spatial information) makes it inherently more complex than images. The final work discussed in this dissertation models the temporal distribution of frame-based descriptors. Chapter 3 showed that modeling the temporal variation of video data can significantly improve the performance of many video understanding tasks. To model the variation of frame-based features in time, we introduced Hard and Soft Cluster Encoding, a novel encoding technique inspired by the Fisher Kernel and VLAD. Results show significant improvements over models which ignore the variation of features in time.

However, our approach has a few limitations. When the number of cluster centroids increases, the video descriptors may contain redundant information due to the increase in the dimensionality of the video descriptors. Our future work plans to investigate the use of an efficient dimensionality reduction technique which does not worsen performance. we also plan to explore additional encoding methods. These studies are made difficult by the lack of large-scale video datasets with temporal variation. Real world videos contain many temporal variations and activities cannot always be labeled frame by frame. Thus, as discussed above, a large-scale realistic dataset with fine-grained labels will be cause significantly accelerated development in video understanding.

While Chapter 2 and Chapter 3 studied daily living activity in a fully supervised setting, chapter 5 discussed methods of performing fine-grained activity recognition using unlabeled data. The task was framed as an MTC problem and a novel algorithm was proposed and evaluated on the Rochester ADL dataset. These results clearly demonstrated the advantage of using an MTC approach (in particular KCEMD-MTC) for ADL analysis. Future works can include ex-

ploiting the proposed MTC algorithms for other vision applications as well as investigating how to improve the current MTC methods (*e.g.* by detecting outlier tasks).

Image classification has had incredible success because of large labeled datasets such as Imagenet. There are also large-scale fine-grained image datasets for various objects such as birds and cars. The lack of such datasets for video understanding has hampered its advancement. Especially in the fine-grained classification scenario, there is often very little labeled data. Recently, Google released Youtube 8 milion [3], a dataset with 8 million youtube videos. However, it does not have fine-grained labels. LSMDC [80], a large dataset for video captioning, is labeled with text captions. Although this dataset contains 120 hours videos, captioning systems require even more labeled training data than classification ones. In order to reach the level of success in videos that has been seen in image based tasks, a large dataset annotated with single word fine-grained labels need to be gathered. However, our work so far has utilized current labeled datasets in conjunction with unlabeled data to move the field forward.

Bibliography

[1] Mojtaba Khomami Abadi, Azad Abad, Ramanathan Subramanian, Negar Rostamzadeh, Elisa Ricci, Jagannadan Varadarajan, and Nicu Sebe. A multi-task learning framework for time-continuous emotion estimation from crowd annotations. In *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*, pages 17–23. ACM, 2014.

[2] A Abella and J R Kender. Qualitatively describing objects using spatial prepositions. *Qualitative Vision, 1993., Proceedings of IEEE Workshop on*, pages 33–38, 1993.

[3] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[4] P. Banerjee and R. Nevatia. Pose filter based hidden-crf models for activity detection. In *ECCV*, 2014.

[5] Hakan Bilen, Vinay P Namboodiri, and Luc. Van Gool. Action recognition: A region based approach. *WACV*, 2011.

[6] Piotr Bilinski and Françoi. Bremond. Contextual statistics of space-time ordered features for human action recognition. *AVSS*, 2012.

[7] Piotr Bilinski, Etienne Corvee, Slawomir Bak, and Francois. Bremond. Relative dense tracklets for human action recognition. *FG*, 2013.

[8] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[9] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[10] Matteo Bregonzio, Shaogang Gong, and Tao Xiang. Recognising action as clouds of space-time interest points. In *CVPR*, 2009.

[11] Patrick Buehler, Mark Everingham, Daniel P Huttenlocher, and Andrew Zisserman. Upper body detection and tracking in extended signing sequences. *International journal of computer vision*, 95(2):180–197, 2011.

[12] Rich Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[13] Tat-Jen Cham and James M Rehg. A multiple hypothesis approach to figure tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[14] James Charles, Tomas Pfister, Derek Magee, David Hogg, Andrew Zisserman, and UK Leeds. Upper body pose estimation with temporal sequential forests. 2014.

[15] M.-Y. Chen, A.G. Hauptmann, and H. Li. Combining motion understanding and keyframe image analysis for broadcast video information extraction. *Evolutionary and Bio-Inspired Computation: Theory and Applications IV, SPIE Defense, Security, and Sensing*, 2010.

[16] Anoop Cherian, Julien Mairal, Karteek Alahari, Cordelia Schmid, et al. Mixing body-part sequences for human pose estimation. In *CVPR 2014-IEEE Conference on Computer Vision & Pattern Recognition*, 2014.

[17] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Int. Workshop on Statistical Learning in Computer Vision*, 2004.

[18] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[19] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[20] Navneet Dalal and Bill. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

[21] Chris HQ Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55, 2010.

[22] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[23] Julian Eggert and Edgar Korner. Sparse coding and nmf. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 2529–2533. IEEE, 2004.

[24] I. Everts, J. van Gemert, and T. Gevers. Evaluation of color stips for human action recognition. In *CVPR*, 2013.

[25] L Fei-Fei and P Perona. A Bayesian hierarchical model for learning natural scene categories. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:524–531, 2005.

[26] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[27] P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[28] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. *CVPR*, 2008.

[29] Katerina Fragkiadaki, Han Hu, and Jianbo Shi. Pose from flow and flow from pose. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2059–2066. IEEE, 2013.

[30] Utkarsh Gaur, Y Zhu, B Song, and A. Roy-Chowdhury. A string of feature graphs model for recognition of complex activities in natural videos. *ICCV*, 2011.

[31] Quanquan Gu and Jie Zhou. Learning the shared subspace for multi-task clustering and transductive transfer classification. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 159–168. IEEE, 2009.

[32] Sung Ju Hwang, Fei Sha, and Kristen Grauman. Sharing features between objects and their attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1761–1768. IEEE, 2011.

[33] B. Ionescu, I. Mironica, K. Seyerlehner, P. Knees, J. Schluter, M. Schedl, H. Cucu, A. Buzo, and P. Lambert. ARF mediaeval 2012: Multimodal video classification. In *MediaEval workshop*, 2012.

[34] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *NIPS*, 1999.

[35] Tommi S Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. *NIPS*, 1999.

[36] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010.

[37] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, PAMI*, 34(9):1704–1716, 2012.

[38] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, December 2013.

[39] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. *BMVC*, 2010.

[40] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012.

[41] Adriana Kovashka and Kristen. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. *CVPR*, 2010.

[42] H. Kuehne, D. Gehrig, T. Schultz, and R. Stiefelhagen. On-line action recognition from sparse feature flow. In *VISAPP*, 2012.

[43] Hildegard Kuehne, Dirk Gehrig, Tanja Schultz, and Rainer. Stiefelhagen. On-line action recognition from sparce feature flow. *VISAPP*, 2012.

[44] Brian Kulis and Michael I Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. *arXiv preprint arXiv:1111.0352*, 2011.

[45] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[46] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin. Rozenfeld. Learning realistic human actions from movies. *CVPR*, 2008.

[47] Jinna Lei, Xiaofeng Ren, and Dieter Fox. Fine-grained kitchen activity recognition using rgb-d. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 208–211. ACM, 2012.

[48] J. Li, S. Gong, and T. Xiang. Scene segmentation for behaviour correlation. *ECCV*, 2008.

[49] Li-Jia Li, Hao Su, Eric P. Xing, and Li Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. *Neural Information Processing Systems (NIPS)*, 2011.

[50] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[51] Z. Lin, Z. Jiang, and L. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009.

[52] K. Liu, M. Weng, C. Tseng, Y. Chuang, and M. Chen. Association and temporal rule mining for post-filtering of semantic concept detection in video. *IEEE TMM*, 2008.

[53] K-H. Liu, M-F. Weng, C-Y. Tseng, Y-Y. Chuang, and M-S. Chen. Association and temporal rule mining for post-filtering of semantic concept detection in video. *IEEE Transactions on Multimedia*, 10(2):240–251, 2008.

[54] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2721, 2013.

[55] Behrooz Mahasseni and Sinisa Todorovic. Latent multitask learning for view-invariant action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3128–3135, 2013.

[56] Manavender R Malgireddy, Ifeoma Nwogu, and Venu. Govindaraju. A generative framework to investigate the underlying patterns in human activities. *ICCV Workshops*, 2011.

[57] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard. YAAFE, an easy to use and efficient audio feature extraction software. In *ISMIR*, 2010.

[58] Pyry Matikainen, Martial Hebert, and Rahul. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. *ECCV*, 2010.

[59] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. *ICCV*, 2009.

[60] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *CVPR*, 2009.

[61] I. Mironica, J. Uijlings, N. Rostamzadeh, B. Ionescu, and N. Sebe. Time matters!: Capturing variation in time in video using fisher kernels. In *ACM Multimedia*, 2013.

[62] Ionut Mironica, Bogdan Ionescu, Jasper Uijlings, and Nicu. Sebe. Fisher kernel based relevance feedback for multimodal video retrieval. *ICMR*, 2013.

[63] Fabian Nater, Helmut Grabner, and Luc Van Gool. Exploiting simple hierarchies for unsupervised human behavior analysis. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2014–2021. IEEE, 2010.

[64] Thuy Thi Nguyen, Nguyen Dang Binh, and Horst Bischof. Efficient boosting-based active learning for specific object detection problems. *Int. Journal of Electrical, Computer, and Systems Engineering*, 3:2070–3813, 2009.

[65] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[66] Sangmin Oh, Anthony Hoogs, Matthew Turek, and Roderic Collins. Content-based retrieval of functional objects in video using scene context. *ECCV*, 2010.

[67] C. Papageorgiou and T. Poggio. A trainable system for object detection. *Int. Journal of Computer Vision (IJCV)*, 38(1):15–33, 2000.

[68] Dennis Park and Deva Ramanan. N-best maximal decoders for part models. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2627–2634. IEEE, 2011.

[69] F. Perronnin, J. Sanchez, and T. Mensink. Improving the Fisher Kernel for Large-Scale Image Classification. In *ECCV*, 2010.

[70] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010.

[71] D. Picard and P-H. Gosselin. Improving image similarity with vectors of locally aggregated tensors. In *ICIP*, 2011.

[72] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[73] G-J Qi, X-S Hua, Y. Rui, J. Tang, T. Mei, M. Wang, and H-J Zhang. Correlative multilabel video annotation with temporal kernels. *ACM TOMCCAP*, 2008.

[74] G-J. Qi, X-S. Hua, Y. Rui, J. Tang, T. Mei, M. Wang, and H-J. Zhang. Correlative multilabel video annotation with temporal kernels. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 5(1), 2008.

[75] D. Ramanan and C. Sminchisescu. Training deformable models for localization. *CVPR*, 2006.

[76] Deva Ramanan, David A Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 271–278. IEEE, 2005.

[77] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *ECCV*, 2010.

[78] K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. In *MVAP*, 2012.

[79] J. Revaud, M. Douze, C. Schmid, and H. Jégou. Event retrieval in large video collections with circulant temporal encoding. In *CVPR*, 2013.

[80] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Chris Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 2017.

[81] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.

[82] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1194–1201. IEEE, 2012.

[83] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. *IEEE Workshop on Applications of Computer Vision*, 2005.

[84] N. Rostamzadeh, G. Zen, I. Mironică, J. Uijlings, and N. Sebe. Daily living activities recognition via efficient high and low level cues combination and fisher kernel representation. In *ICIAP*, 2013.

[85] Negar Rostamzadeh, Jasper Uijlings, Ionuj Mironică, Mojtaba Khomami Abadi, Bogdan Ionescu, and Nicu Sebe. Cluster encoding for modelling temporal variation in video. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3640–3644. IEEE, 2015.

[86] Negar Rostamzadeh, Jasper Uijlings, and Nicu Sebe. Action recognition using accelerated local descriptors and temporal variation.

[87] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998.

[88] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[89] Sreemanananth Sadanand and Jason J. Corso. Action bank: A high-level representation of activity in video. *CVPR*, 2012.

[90] Ruslan Salakhutdinov, Antonio Torralba, and Josh Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Com-*

puter Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 1481–1488. IEEE, 2011.

[91] Ben Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3674–3681. IEEE, 2013.

[92] Benjamin Sapp, Alexander Toshev, and Ben Taskar. Cascaded models for articulated pose estimation. In *Computer Vision–ECCV 2010*, pages 406–420. Springer, 2010.

[93] Benjamin Sapp, David Weiss, and Ben Taskar. Parsing human motion with stretchable models. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1281–1288. IEEE, 2011.

[94] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *ECCV*, 2010.

[95] Silvio Savarese, Andrey DelPozo, Juan Carlos Niebles, and Li. Fei-Fei. Spatial-temporal correlations for unsupervised action classification. *IEEE Workshop on Motion and Video Computing*, 2008.

[96] S. Schmiedeke, P. Kelm, and T. Sikora. TUB @ MediaEval 2012 tagging task: Feature selection methods for bag-of- (visual)-words approaches. In *MediaEval Workshop*, 2012.

[97] S. Schmiedeke, C. Kofler, and I. Ferrané. Overview of MediaEval 2012 genre tagging task. In *MediaEval workshop*, 2012.

[98] S. Schmiedeke, P. Xu, I. Ferrané, M. Eskevich, C. Kofler, M. Larson, Y. Estève, L. Lamel, G. Jones, and T. Sikora. Blip10000: A social video dataset containing spug content for tagging and retrieval. In *ACM Multimedia Systems*, 2013.

[99] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICIP*, 2004.

[100] T. Semela, M. Tapaswi, H. Ekenel, and R. Stiefelhagen. Kit at mediaeval 2012 - contentbased genre classification with visual cues. In *MediaEval workshop*, 2012.

[101] Horesh Ben Shitrit, Jerome Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. *ICCV*, 2011.

[102] Hedvig Sidenbladh, Michael Black, and David Fleet. Stochastic tracking of 3d human figures using 2d image motion. *Computer VisionECCV 2000*, pages 702–718, 2000.

[103] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[104] Cristian Sminchisescu and Allan Jepson. Variational mixture smoothing for non-linear dynamical systems. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[105] C. Snoek and M. Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2008.

[106] Cees GM Snoek and Marcel Worring. Multimedia event-based video indexing using time intervals. *Transactions on Multimedia*, 2005.

[107] C.G.M. Snoek and M. Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2009.

[108] B. Solmaz, S. Assari, and M. Shah. Classifying web videos using a global video descriptor. *Machine Vision and Applications*, 2012.

[109] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *CVPR*, 1999.

[110] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. *CVPR*, 1999.

[111] J. Stöttinger, A. Hanbury, N. Sebe, and T. Gevers. Sparse color interest points for image retrieval and object categorization. *TIP*, 2012.

[112] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Human activity detection from rgbd images. *plan, activity, and intent recognition*, 64, 2011.

[113] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *International Conference on Pervasive Computing*, pages 158–175. Springer, 2004.

[114] C. Tomasi and T. Kanade. Detection and tracking of point features. *Technical Report CMU-CS-91-132, Carnegie Mellon University*, 1991.

[115] Carlo Tomasi and Takeo. Kanade. Detection and tracking of point features. Technical report, CMU-CS, 1991.

[116] D. Tran and D. Forsyth. Improved human parsing with a full relational model. *ECCV*, 2010.

[117] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[118] Matthew Turek, Anthony Hoogs, and Roderic Collins. Unsupervised learning of functional categories in video scenes. *ECCV*, 2010.

[119] J. Uijlings, I. Duta, N. Rostamzadeh, and N. Sebe. Realtime video classification using dense HOF/HOG. In *ICMR*, 2014.

[120] Jasper RR Uijlings, IC Duta, Negar Rostamzadeh, and Nicu Sebe. Realtime video classification using dense hof/hog. In *Proceedings of International Conference on Multimedia Retrieval*, page 145. ACM, 2014.

[121] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *TIP*, 2009.

[122] J. van Gemert, J-M. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, 2008.

[123] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[124] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *CVPR*, 2001.

[125] Paul Viola, Michael Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *ICCV*, 2003.

[126] G Wang and D Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *International Conference on Computer Vision*, 2009.

[127] H. Wang, A. Kläser, C. Schmid, and C. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103:60–79, 2013.

[128] H. Wang, A. Kläser, C. Schmid, and C-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[129] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[130] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013.

[131] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia. Schmid. Evaluation of local spatio-temporal features for action recognition. *BMVC*, 2009.

[132] J. Wang, Z. Chen, and Y. Wu. Action recognition with multiscale spatio-temporal contexts. In *CVPR*, 2011.

[133] Meng Wang and Xiaogang Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. *CVPR*, 2011.

[134] Xiaogang Wang, Cha Zhang, and Zhengyou Zhang. Boosted multi-task learning for face verification with applications to web image and video search. In *computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on*, pages 142–149. IEEE, 2009.

[135] D. Weiss and B. Taskar. SCALPEL: Segmentation cascades with localized priors and efficient learning. In *CVPR*, 2013.

[136] Geert Willems, Tinne Tuytelaars, and Luc. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. *ECCV*, 2008.

[137] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.

[138] Yan Yan, Elisa Ricci, Negar Rostamzadeh, and Nicu Sebe. It's all about habits: Exploiting multi-task clustering for activities of daily living analysis. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 1071–1075. IEEE, 2014.

[139] Yan Yan, Elisa Ricci, Ramanathan Subramanian, Gaowen Liu, and Nicu Sebe. Multitask linear discriminant analysis for view invariant action recognition. *IEEE Transactions on Image Processing*, 23(12):5599–5611, 2014.

[140] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[141] Yi Yang and Deva. Ramanan. Articulated human detection with flexible mixtures-of-parts. *PAMI*, 2012.

[142] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2878–2890, 2013.

[143] Angela Yao, Juergen Gall, and Luc Van Gool. Coupled action recognition and pose estimation from multiple views. *International journal of computer vision*, 100(1):16–37, 2012.

[144] Chunfeng Yuan, Weiming Hu, Guodong Tian, Shuang Yang, and Haoran Wang. Multi-task sparse learning with beta process prior for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 423–429, 2013.

[145] Gloria Zen and Elisa Ricci. Earth mover's prototypes: A convex learning approach for discovering activity patterns in dynamic scenes. *CVPR*, 2011.

[146] Gloria Zen, Negar Rostamzadeh, Jacopo Staiano, Elisa Ricci, and Nicu. Sebe. Enhanced semantic descriptors for functional scene categorization. *ICPR*, 2012.

[147] Jianwen Zhang and Changshui Zhang. Multitask bregman clustering. *Neurocomputing*, 74(10):1720–1734, 2011.

[148] Yimeng Zhang, Xiaoming Liu, Ming-Ching Chang, Weina Ge, and Tsuhan. Chen. Spatio-temporal phrases for activity recognition. *ECCV*, 2012.

[149] Zhenfeng Zhu, Hanqing Lu, James Hu, and Keiichi Uchimura. Car detection based on multi-cues integration. *International Conference of Pattern Recognition (ICPR)*, 2004.

[150] Silvia Zuffi, Javier Romero, Cordelia Schmid, and Michael J Black. Estimating human pose with flowing puppets. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3312–3319. IEEE, 2013.