Human Action Recognition from Gradient Boundary Histograms

by

Xuelu Wang

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.SC. degree in
Computer Science

uOttawa

School of Electrical Engineering and Computer Science
Faculty of Engineering, University of Ottawa
Ottawa, Canada

Abstract

This thesis presents a framework for automatic recognition of human actions in uncontrolled, realistic video data with fixed cameras, such as surveillance videos. In this thesis, we divide human action recognition into three steps: description, representation, and classification of local spatio-temporal features. The bag-of-features model was used to build the classifier. Fisher Vectors were also studied. We focus on the potential of the methods, with the joint optimization of two constraints: the classification precision and its efficiency.

On the performance side, a new local descriptor, called Gradient Boundary Histograms (GBH), is adopted. It is built on simple spatio-temporal gradients, which can be computed quickly. We demonstrate that GBH can better represent local structure and motion than other gradient-based descriptors, and significantly outperforms them on large datasets. Our evaluation shows that compared to HOG descriptors, which are based solely on spatial gradient, GBH descriptor preserves the recognition precision even in difficult situation.

Since surveillance video captured with fixed cameras is the emphasis of our study, removing the background before action recognition is helpful for improving efficiency. We first preprocess the video data by applying HOG to detect humans. GBH descriptor is then used at reduced spatial resolutions, which yields both high efficiency and low memory usage; in addition, we apply PCA to reduce the feature dimensions, which results in fast matching and an accelerated classification process.

Experiments our methods achieved good performance in recognizing precision, while simultaneously highlighting effectiveness and efficiency.

Acknowledgements

I would like to express my sincerest appreciation and thanks to my supervisor, Professor Robert Laganière, for his guidance, support and patience during my graduate experience. I would like to thank him for encouraging my research, giving me useful remarks and offering me help in the research of this master thesis. His trust in my abilities and the academic latitudes he provided have proved to be extremely invaluable during my M.A.Sc. study. He has provided far more than just the required supervisory feedback and excellent guidance throughout the journey of the research and the production of this work.

I would like to thank all the members in Viva Lab for their thoughtful comments on my thesis. Sincere thanks go to Dr. Andres Solis Montér for the invaluable assistance and guidance provided throughout my research. I also thank Dongfeng Gu and Muye Jiang, who supplied considerable suggestions and help with my research.

I would also like to thank Jun Yang, who has been not only a wise mentor but also a good friend to me. He has provided me extensive advice and assistance during my studies and research at the University of Ottawa.

I would like to thank my family and Fei Dong, who have supported me throughout the entire process of striving towards my goal in Canada. This work is dedicated to them.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Histogram of Oriented Gradients (HOG)

A feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. It takes advantage of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, and is computed on a dense grid of uniformly spaced cells.

Gradient Boundary Histogram (GBH)

A local descriptor based on pure spatio-temporal gradients, which are computed by applying simple 1-D [-1,0,1] Sobel masks on both x and y directions, followed by a [-1, 1] temporal filter over two consecutive gradient images.

Non-Maximum Suppression (NMS)

In object detection, it is used to transform a smooth response map that triggers many imprecise object window hypotheses in, ideally, a single bounding-box for each detected object.

Principal Component Analysis (PCA)

According to Wikipedia, "A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables."

Bag-of-Feature (BoF)

Also known as bag-of-words, which is originally applied to document analysis. It treats visual features as words and has been widely used in image/video classification. A bag of visual features is represented by a vector of occurrence counts of a vocabulary of local image/video features in computer vision.

Fisher Vector (FV)

A statistics encoding method which describes the distribution of a set of local im-

age/video descriptors. FV applies Gaussian Mixture Model (GMM) and extends the BoF by encoding high-order statistics (first and, optionally, second order) between the descriptors. As a consequence, both codewords' occurrences and additional information about the distribution of the descriptors are encoded at the same time.

Support Vector Machines (SVMs)

According to Wikipedia, "SVMs are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier."

One-verse-One (OVO)

Also called one-against-one (OAO), a method to perform multi-class classification by decomposing the multi-class problem into a number of two-class problems, and applying binary classification for each of them. For n classes, OVO constructs $n(n1)/2$ SVM classifiers. The SVM models are learned using data from any two of the classes. After all models are trained, the prediction is implemented by max-wins voting, with class labels assigned to the class with the largest vote from the models.

One-verse-All (OVA)

Also called one-against-all (OAA), a method to perform multi-class classification by decomposing the multi-class problem into a number of two-class problems, and applying binary classification for each of them. For n classes, OVA constructs n SVM models. The ith SVM is trained with all of the examples from the ith class as positive labels, and remaining examples from other classes as negative labels.

Chapter 1

Introduction

Computer vision, as a discipline of science, has received a significant amount of attention in recent years. It refers to acquiring better perceptions of the environment from computing systems. One prominent task of these recent advances in computer vision is in the field of action recognition in video, which has been a very active field of research over the past few years. Action recognition aims at identifying human actions. It is considered one of the essential premises for the analysis of human behaviors from video data.

The rapid spread of surveillance cameras has led to a wide use of video surveillance in daily life. A number of technologies have been studied that detect specific human actions by analyzing image features in video sequences of surveillance cameras. With the

ubiquity of video surveillance and reliable, automatic recognition technology, efficiency and security can be improved. Surveillance cameras in fast-food restaurants guarantee security, and provide statistical feedback of different activities happening around the dining and ordering areas. For example, with the collected behavior information from the ordering area, managers can adjust the existing arrangements and operations.

1.1 Problem Statement

For this thesis, we focus on real-time automatic recognition of human actions for uncontrolled, realistic video data, such as surveillance videos. The current situation on video action recognition is that most of the improvements targets more on recognition precision, but less on efficiency. On the contrary, efficiency is our highest priority, as the object of our thesis is to implement recognition in real-time.

Local spatio-temporal features and bag-of-features(BoF) representations [10] [22] [48] have recently become popular for action recognition due to their simplicity and good performance. All these methods achieve impressive progress, but there are still some problems that need to be addressed.

In terms of descriptor, the most popular one based on designed feature method is based on three-dimensional (3D) gradients [20] [40]. It encodes the spatio-temporal information, and stabilize to rotation, scale and brightness variation. However, three gradient components are used to build descriptors, which leads to high dimensionality and relatively expensive quantization cost.

Additionally, most existing action recognition methods use sparse, interest-point representations, resulting in loss of data, and a failure to transmit sufficient information to the classification step. Dense sampling methods are, therefore, becoming increasingly popular, since they can provide a very large number of feature patches, and can then achieve excellent performance; however, a computationally expensive feature extraction process is used in those action recognition methods, and the increase of the number of processed points in dense sampling methods adds to the computation complexity.

Finally, accumulative approaches such as Bag of Feature also causes the loss of tempo-

ral information and global spatial structure, and only the statistics of unordered primitive features from the image sequences are captured. A more discriminating method should include global structure information and ordering of local events.

1.2 Motivation of the Problem

We focus on solutions to those challenges, and aim at achieving the real-time recognition by improving efficiency.

Firstly, a more efficient strategy is used to reduce memory usage. For the videos captured with fixed cameras, background information is most often not meaningful. We propose to remove it before recognition. This way, only the most important part of the video will be processed.

Secondly, We intend to encode both local static appearance and motion information. However, we want to avoid using three gradient components. In uncontrolled, realistic video data, human performance could be relatively complex. The requirement for local spatio-temporal features is then higher. It should supply rich motion information to build precise feature for recognition precision, and the achievement should not be the cost of efficiency.

1.3 Contributions

This work makes several contributions showing the efficiency and effectiveness of the proposed methods:

- We implement the real-time recognition by applying a new local 3D descriptor based on a histogram of oriented ST gradients.
- A new method is proposed, which is based on removing the useless information before analyzing the actions. This method speeds up the process of building features by saving processing time.

- We evaluate two methods of representation, and compared different combinations with linear and non-linear SVM classifications.

1.4 Thesis Outline

The thesis is organized as follows:

- Chapter 2 presents the background and related works of this research. The detailed procedures are introduced first. The previously published approaches are also stated in this chapter. As the highly related part to estimate the results, this chapter also presents widely used human action datasets.

- By building the descriptor, Chapter 3 introduces our approach to visualize the action feature from videos. One of the main focuses of this thesis, Gradient Boundary Histogram descriptors, is introduced in this chapter. To obtain the better descriptors, Chapter 3 also illustrates the popular strategies, such as non-maximum suppression and PCA. Lastly, the final conclusion are presented.

- In Chapter 4, feature representation and classification are introduced, respectively. As the most two popular choices, the BoF approach and the FV method are applied to all the descriptors to extract features, the specific procedure is presented in Chapter 4. SVM is then illustrated as the method of classification. The overall estimation is summarized at the end of this chapter.

- In Chapter 5, an evaluation of performance of our algorithm is presented. By using different metrics, our evaluation on KTH dataset shows that our approach is able to obtain considerable high precision and efficiency.

- Chapter 6 concludes the thesis and highlights the contributions of this research. Future works are also discussed.

Chapter 2

Related Work

In this chapter, state-of-the-art approaches of action recognition are introduced to recognize human action in realistic, uncontrolled videos. As an overall acknowledge, the entire process of human action recognition is described in the first place. We then review previously published methods on efficient action recognition. This chapter ends with the brief introduction of certain popular released human action datasets.

2.1 Methods based on Designed Descriptors

Globally, the process of human action recognition can be summarized as feature representation and classification. In this section, each procedure is described in detail, following

by the overviews of different approaches.

2.1.1 Representation

Feature representation is considered as the key step in action recognition. It distinguishes movement and extracts distinctive features from video data. Feature representation is generated by combining a series of feature descriptors in a video. Different methods exist for the implementation of representation. We divide them into three categories: body modelling methods; global representation methods; local feature methods.

2.1.1.1 Body modelling methods

Body modelling methods use human anatomy as a direct foundation and the process of action recognition using this method can be summarized as follows. Firstly, 2D motion patterns or 3D body models for human body parts are generated, and a body model is used to detect the various body parts. The motion of the body model is then extracted from the video data, and compared to the ground truth body models, which have already been constructed either globally or through the use of makers to capture the human performance in specific environments. Released approaches can be divided into two subsets: direct recognition and recognition by reconstruction.

Direct recognition employs 2D motion patterns; various investigations take advantage of 2D models of anatomical landmarks, such as joint positions. Johansson [17] is the most representative pioneer in this area; his contribution reflects in what is referred to as biological motion analysis, that is the visual perception of motion patterns features of moving live beings. Human action recognition could be made in such way that a certain amount of Moving Light Displays (MLD) attached to the human body can be detected.

Recognition by reconstruction detects stick figures in space-time volume. It is a 3D approach which computes 3D model based on the captured motion of the body. Considering the great variation in human bodies with different degrees of freedoms, this process is quiet challenging. There are two ways to implement this method; the first is called top-down approach, which employs a geometrical model. 3D body models are obtained from a sequence of images by matching 2D samples such as joint angle

parameters. A method is proposed by Marr and Nishihara [26] which includes cylindrical primitives. The dual method is known as bottom-up approach,and depends on low-level image features. 2D body parts are tracked using rectangular patches and 3D models are determined by upgrading 2D using an orthographic camera model[35].

For simpler and less complicated datasets, body modelling methods perform comparatively well. However,it should be noted that the detection of human body parts and human poses could impact the result of those methods. Moreover, the techniques that are used to implement the detection dynamically prove challenging, and some of them remain open and active, such as human tracking, 3D reconstruction, and image/video segmentation.

The performance of most body modeling methods is neither robust nor reliable for uncontrolled, realistic video data. Because model-based approaches are sensitive to background clutter, noise and occlusions, scale change and multiple subjects etc., it is very difficult to successfully extract high-level representations from images.

2.1.1.2 Global representation methods

Global representation methods use the entire human body to generate the observations. They employ a more efficient and robust method by taking advantage of the entire model's global appearances and motions, instead of detecting and labelling individual body parts. In this section, details of this approach are explained from three aspects: Shape-based representation, Motion-based representation, and a Combination of Shape and Motion-based representation.

Shape-based representation In shape-based representation, the silhouette or the contour of actors is encoded in various ways. The fundamental idea behind shape-based representation is that an action in the video could be regarded as the division of a sequence of single frames, and each frame contains a pose which can be detected and encoded using shape features. Similarly, single frame recognition is able to be extended to multiple frames for robust action recognition. However, shape-based representations are heavily sensitive to background segmentation, which is ubiquitously unstable in realistic

situations.

Yamato et al. [52] utilize silhouette images by separating them into a grid of cells, and determine mesh features by computing the ratio of foreground to background pixels from each cell. They apply hidden Markov models (HMM) to the processed feature sequences to implement the recognition.

Wang and Suter [49] employed two computational representations: average motion energy (AME) and mean motion shape (MMS) to display the human silhouettes extracted from videos. They use a supervised pattern classification method by using various distance measurements to achieve classification. Given a sequence of binary silhouette images involving a moving human, the AME is computed from periodical motion detection, whereas the MMS is computed based on the boundary or shape obtained from single-connectivity binary silhouette using a border.

Weinland and Boyer [50] represent motion sequences through a set of time-invariant static key-pose exemplars. To represent the actions, distances between silhouettes exemplars and the frames in the action's sequence are computed, and a set of minimum distances for each exemplar are gathered to be a vector representation. A Bayes classifier is then used in the final step.

To obtain a deeper understanding, the appearance over a long sequence of video frames is studied. By stacking multiple silhouette images into volumetric representation, a temporal template is built. Bobick and Davis [5] extracted silhouettes from a single view and aggregated differences between subsequent frames of an action sequence. They employed a binary motion energy image (MEI) to identify the region of motion, which is a single image gathered in successive silhouette frames. In addition, another concept known as motion history image (MHI) is proposed, that indicates the relationship between the history of that pixel over time.

Recently, silhouette-based representation was extended to three-dimensional volume. It is achieved by stacking single silhouettes detected in each frame, and expressing the whole body appearance as a function of space and time. The first attempt of this idea is proposed by Blank et al. and Gorelick et al. [4]. To generate space-time shapes, they stacked a sequence of silhouette images, which are computed through background

subtraction. They also applied the properties of the solution to the Poisson equation to extract salient space-time features, whose weighted moments are calculated and described by a high-dimensional feature vector. These global vectors are then matched to space-time shapes in test sequences for classification.

Motion-based representation Motion information can also be used for action recognition on top of shape information. Optical flow, as the principal manner for describing kinematics, exploit pixel-wise oriented differences between subsequent frames. It performs very well when background subtraction cannot be implemented. However, dynamic backgrounds and camera movement add noise in the motion descriptor.

Polana and Nelson [34] determined a region of interest (ROI) over the person, and accumulated flow magnitudes in a regular grid of non-overlapping bins by computing the optical flow of ROI. The descriptors of test sequences are matched to the reference motion templates in the classification.

Efros et al. [11] utilized optical flow to recognize human actions at a distance. They used a camera operator to keep the moving figure in the centre of the field of view and capture small figure-centric sequences. The optical flow vector field computed for each frame was first split into two scalar fields corresponding to the horizontal and vertical components of the flow, and each field is then half-wave rectified into four non-negative channels. To avoid noisy displacement for getting the final motion descriptor for each frame, they also employed the blurred and normalized processes. The four channels are matched separately to a test sequence sequence, which is aligned frame-wise to a database of annotated actions to achieve classification.

Ali and Shah [1] derived kinematic features from the optical flow, giving rise to a spatio-temporal pattern. Dominant kinematic trends or kinematic modes are then captured by performing Principal Component Analysis (PCA) on the spatio-temporal volumes of the kinematic features. For classification, the authors propose the use of multiple instances learning (MIL), in which each action video is represented by a bag of kinematic modes. Each video is then embedded into a kinematic mode-based feature space,and the coordinates of the video in that space are used for classification using the

nearest neighbour algorithm.

The concept of template can also be combined with optical flow, and in this situation, templates are computed by applying spatio-temporal volume filters. Ke et al. [19] utilized this idea to generalize the 3D spatio- temporal volumetric features. First, they extracted the feature by dividing the optical flow into horizontal and vertical components; volumetric features on each component were computed and represented by one-box or two-box volumes. In this process, they also applied "integral video" data structure to reduce computing time. They then use a sliding-window to select a small subset of volumetric features and arranged them in a cascade; this method is called direct forward feature selection method, and could achieve action detection efficiently.

Shape and Motion-Based Representation To achieve better recognition, information of both shape and motion is utilized simultaneously. Recently, a deep extension has been processed where space–time volumes of silhouettes and flow are combined into implementation. In this way, all aspects of information could be captured. Moreover, the representation is continuous and there is no need of time warping.

Yilmaz and Shah [54] propose to perform action recognition view invariantly. The authors firstly extracted 2D contours in the image plane, the point on which the outer boundary of the object are projected, and extend them to a spatiotemporal volume with respect to time. The action sketch is then computed by analyzing the spatiotemporal volume with differential geometric surface properties, which is related to various types of motions and object deformations also invariant to the viewing angle of the camera. They posed the matching problem in terms of matching a set of points in one view of an object, while a set of points in another view.

Yan et al.[53] employed the concept of action sketch,and extended it to a 4D action feature model (4D-AFM) for representation and recognition of actions from arbitrary views. They first constructed 3D exemplars from multiple views for each frame in a training sequence. An action sketch was then calculated for each view based on the spatio-temporal volumes of this view, and projected onto the constructed 3D exemplars. The 4D-AFM was built by mapping action sketches to the sequence of 3D visual hulls over

time. The authors [53] matched action features from the input videos to the model points of 4D-AFMs by exploiting pairwise interactions of features to achieve action recognition.

2.1.1.3 Local representation methods

Local representation methods describe the action in a video as a collection of local descriptors or patches, instead of one global descriptor as in the global methods. In addition, the features are usually extracted directly from video, which means that there is no need for pre-processing steps such as background subtraction. In comparison, local representations have a remarkable performance against the captured conditions due to their invariance to the representation of events.The framework of local representation consists of three main components: feature detector, feature descriptor, and feature representation. Each part of this framework will be explained in this section.

Feature detectors Feature detectors usually select spatio-temporal locations that contain most distinctive information for the recognition of human action. Those locations are called space–time interest points; their local neighbourhoods vary significantly in both spatial and temporal domains.

Laptev and Lindeberg [22] put forward space–time interest points for the first time, and proposed the Harris3D detector building on 2D Harris-Laplace detector. The detection is achieved by computing a spatio-temporal second-moment matrix at each video point with independent spatial and temporal scale values. They also proposed an optional mechanism for selecting different spatio-temporal scales. They also used local maxima to demonstrate the final locations of space-time interest points.

Typically, Harris3D detector can only detect a small number of features. Dollár et al. [10] addressed this problem by proposing cuboid detector, which applies Gabor filtering in the temporal dimension to select local maxima over space and time of the response function. However, the size of these cuboids is determined by the user, and the number of interest points is influenced by the size of the cuboids, which means that their features are not scale-invariant.

To overcome the problem of scale-invariance, Willems et al. [51] proposed the Hes-

sian3D detector and an extended SURF descriptor. They employed the 3D Hessian matrix to detect salient points that can automatically find the position and scale of the points. The density of features can vary from very sparse to very dense by changing the threshold and the effect on the computation time is minimal. As an enhanced version, not only did they succeed in detecting relative denser space-time points, but they also managed to make this process more computational and efficient by determining the position and scale of the interest points simultaneously, without the time-consuming repeated procedures.

As researchers gradually increase their requirements and expectations, the drawback of sparse interest point detectors becomes increasingly obvious. On one hand, they are computationally expensive when the amount of processed data is large; on the other hand, they cannot acquire enough relevant information for classification, because they miss important aspects in some circumstances. To overcome this difficulty, dense sampling methods are introduced, which sample every pixel in each spatial scale and capture almost all relevant information.

Wang et al. [48] evaluated the performance of three space-time interest point detectors, Harris3D, Cuboid and Hessian3D, comparing their results with dense sampling in a common experimental setup and same bag-of-features SVM recognition framework. The experiences demonstrated that dense sampling at regular space-time grids showed excellent performance when compared to state-of-the-art interest point detectors.

The methods based on interest point detectors are designed to simply integrate 2D space domain and 1D time domain into a joint 3D space, which is not necessarily the best approach since they possess very different characteristics. Some recent works have attempted to operate them in a different manner, by obtaining the relationships between local space-time features through the tracking of interest points in video sequences.

Messing et al. [29] obtained feature trajectories by tracking Harris3D interest points using the KLT tracker. They applied quantized velocity in log-polar coordinates to represent feature trajectories by varying length. Also, a generative mixture of Markov chain models was applied for action recognition.

Conversely, Matikainen et al. [28] used the standard KLT tracker to represent feature

trajectories with fixed length. They cluster the trajectories in the video by K-means and generate an affine transformation matrix for each cluster centre. The elements of those matrices are used to represent the final trajectories which, therefore, also contain information about the displacement vectors.

The trajectories we mentioned previously are often obtained by tracking sparse interest points using the KLT tracker. Since we have already discussed superiorities of the dense sampling, it is obviously a better choice to obtain the trajectories by tracking densely sampled points.

Wang et al. [47] achieved the extraction of dense trajectories using optical flow fields. Feature points were sampled densely on a grid spaced in multiple spatial scales and tracked by employing a state-of-the-art optical flow algorithm separately at each scale. The local motion and appearance around a trajectory are described by the HOG, HOF and MBH to build the final dense trajectories for action recognition. Dense trajectories not only contain enough motion information, but also are robust to fast irregular motions and shot boundaries.

Feature descriptors Feature descriptors are computed to characterize the space-time structure around interest points by image measurements, such as spatial or spatio-temporal image gradients and optical flow, which could emphasize the parts in the video where changes occur. In order to achieve robust action recognition, the descriptor should be distinctive and insensitive to local image deformations.

Dollár et al. [10] experimented normalized pixel values, brightness gradient and windowed optical flow. They concatenated the gradients computed for each pixel in the patch into a single vector to build up the descriptor. Also, principal component analysis (PCA) is employed on the training samples to bring the feature vectors to a lower dimensional space.

Laptev et al. [23] proposed the HOG/HOF descriptors by combining histograms of oriented spatial gradients (HOG) and histograms of optical flow (HOF) to capture the information of local motion and appearance at the same time. They computed gradients and dense optical flow first. Each local region is subdivided into a grid of N * N

* M cells. For each cell, 4-bin HOG histograms and a 5-bin HOF histogram are computed. Normalized histograms are concatenated into HOG, HOF as well as HOG/HOF descriptor vectors, and are similar in spirit to the well-known SIFT descriptor. The addition of HOF brings significant improvements, because the gradient direction of HOG is two-dimensional, resulting in the lack of temporal information in the video.

HOG3D, proposed by Kläser et al. [20], changed the underlying approach by considering the three-dimensional gradient instead of the 2D gradient. A 3D patch is divided into a grid of cells, which are then divided into several separate sub-blocks. A mean 3D gradient for each sub-block is computed and uniformly quantized by assembling regular polyhedrons. Finally, for the given 3D patch, the 3D histogram of oriented gradients is generated by concatenating gradient histogram of all cells, and normalization by the L2 norm.

Wang et al. [47] employed MBH descriptor to represent resulting dense trajectories for action recognition. MBH descriptor was introduced by Dalal et al. [9], who divided the optical flow into two components along horizontal and vertical directions, and computed the spatial derivatives for each of them. Orientation information was then quantized into histograms, and the magnitude is processed for weighting. In this way, camera motion was efficiently reduced, which makes MBH descriptor outperform other state-of-the-art descriptors on human action classification.

Feature representation The purpose of feature representation is to describe a video into a unified form that is suitable for classification. The basic approach is that given a set of extracted local patch descriptors, we encode them into a high dimensional vector and pool them into an video-level signature. Quantizing the local descriptors into a finite set of prototypical elements is the most common patch encoding strategy, which is also the main idea of Bag-of-Feature representation. However, as an alternative patch encoding strategy, Fisher Vector (FV) describes patches by their deviation from an "universal" generative Gaussian mixture model based on fisher kernel. These two solutions are explained here.

Schuldt et al. [39] proposed bag-of-features which originates from the bag-of-words

representation. As the name implies, bag-of-words representation takes advantage of words to represent text, based on the number of times a word appears in the text, while ignoring the order and grammar. All the words appearing in the text are put into one bag, and taken out of order, then placed in the corresponding histogram bins, which are also called vocabulary. The histograms record the number of times different words appear in different texts, which tends to be more useful than a direct word-by-word comparison.

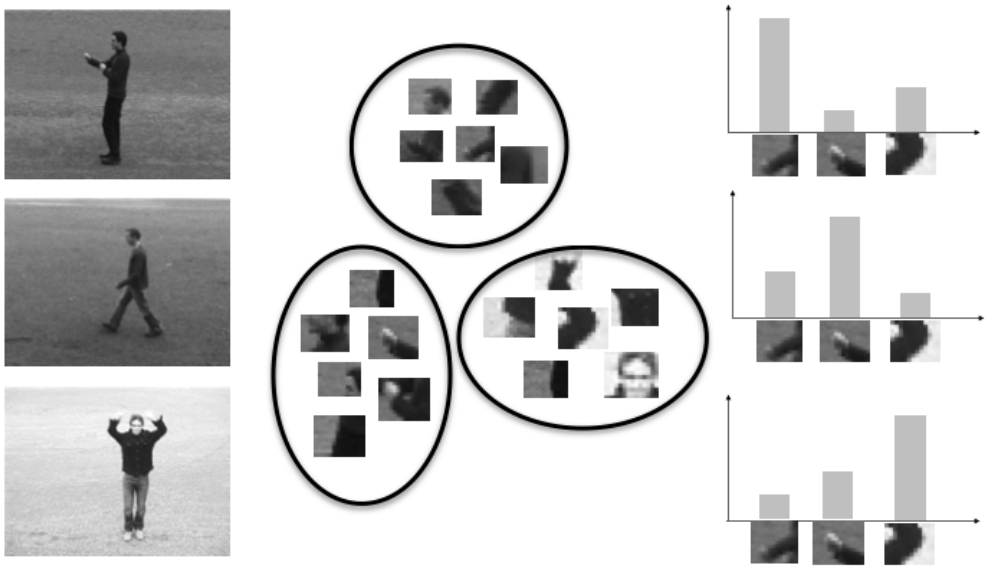

Figure 2.1: Bag-of-features representation.

Bag-of-features (BoF) is extended from bag-of-words for action recognition to represent a video. By applying feature extraction, a video is modelled as a set of local spatio-temporal patches, which are then represented as a single feature vector by the feature descriptors. The vocabulary is now defined based on the feature vectors computed from training data by applying a clustering algorithm, such as K-means, on those feature vectors. The histogram is generated by quantizing the feature vectors into their closest bin. Finally, normalization is applied to make all feature vectors comparable. The representation is summarized in Figure 2.1. Bag-of-feature shows great popularity in object classification from images and action recognition from video data, due to its simplicity and good performance.

BoF is simple to achieve because it makes all the features orderless; however, it loses the spatial and temporal information at the same time. It may cause classification prob-

lems as confusion between similar classes. Many works successfully import geometry and temporal information to avoid this situation. Hamid et al. [13] proposed an unsupervised method for detecting anomalous activities by using bags of event n-grams.

Herve et al. [16] introduced a representation called Fisher Vector, which is a powerful tool to transform an incoming variable-size set of independent samples into a fixed-size vector representation. Based on Fisher Kernel, Fisher Vector encodes high-order statistics between the video descriptors and a diagonal covariance Gaussian mixture model (GMM). This description vector is built by computing the gradient of the sample's probability through the parameters of this distribution, then scaling by the inverse square root of the Fisher information matrix. As a result, it gives the direction, in parameter space, into which the learned distribution should be modified to better fit the observed data. In other terms, FV describes how the set of descriptors deviates from an average distribution of descriptors, modeled by a parametric generative model. The FV is seen as a method to capture the information conveyed by a set of descriptors into a fixed-length signature.

Sanchez et al. [38] proved that Fisher Vector could give superior performance in image classification. The authors proposed some advantages about FV representation by analyzing the results of experiments. In terms of accuracy, the additional gradients incorporated in the FV and the more general way to define a kernel from a generative process of the data offer the doubling guarantees. In terms of to efficiency, not only can it be computed from much smaller vocabularies, but also performs well with simple linear classifiers. As a consequence of that, there is a significant decrease on the cost of computing the representations, the cost of learning the classifiers on these representations as well as the cost of classifying a new image. Wang et al. [46] applied Fisher vectors to encode local descriptors into a holistic representation and proved that the same conclusion also holds for a variety of recognition tasks in the video domain.

2.1.2 Classification

In pattern human action recognition, classification is the last step to achieve the final result. The classification can be unsupervised or supervised.

2.1.2.1 Unsupervised Classification

In the unsupervised approach, we have no information about the labels or constraints between the data. The videos are grouped into several clusters based on their similarities. The number of clusters can be fixed or variable, contributing to the problem of determining how many clusters to choose. Unsupervised learning is simple and efficient. However, evaluation of the results remains a challenge. A typical example of unsupervised learning is K-means clustering. This is a clustering method which aims at partitioning n observations into k clusters, and each observation belongs to the cluster with the nearest mean.

2.1.2.2 Supervised Classification

Supervised classification uses supervised training data to train a classifier that encodes the differences between the classes. "Supervised" here means the labels of the samples are known, so the classifier can learn the relationship between the features and the labels. Usually, the supervised approaches need significant amounts of training data, and the accuracy of the classifier is questionable if the data underfits to the training data. The most famous one is Support Vector Machine (SVM) Figure 2.2 shown a simple situation of SVM.

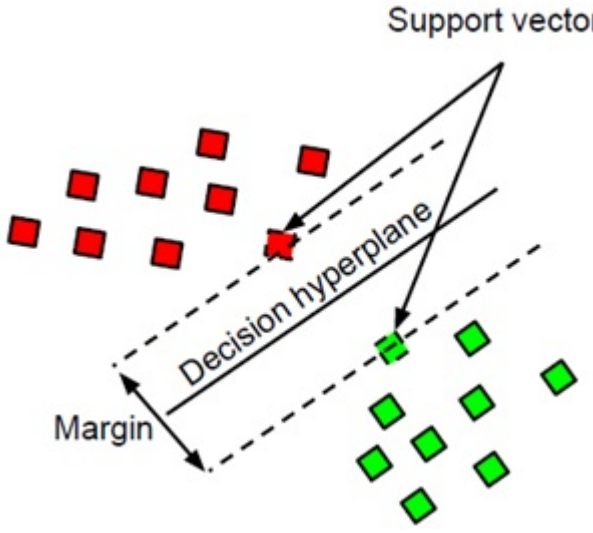

Figure 2.2: Support Vector Machine.

2.2 Deep models Human action recognition

In all the state-of-the-art methods for human action recognition that we discussed above, the features are called hand-crafted features as they are designed to be optimal for different task.

In machine learning tasks, the approaches known as deep models have taken to the stage. They could learn multiple layers of feature hierarchies and automatically built high-level representations from the raw input. Due to the completely automated feature construction, they tended to be more generic.

2.2.1 Convolutional Neural Network

The most popular deep model is the Convolutional Neural Network(CNN)architecture. It is a bioinspired hierarchical multilayered neural network and takes advantage of the 2D structure of an input image to learns visual information directly from the pixels without any pre-processing step. As CNNs develops rapidly, a small number (typically less than 15) of successive frames can be taken as input, but only short sub-sequences assigned as a vector of features with a label are used for training instead of the entire sequence. Proceeding this way, long term temporal information is ignored. CNNs have been proved successful for image processing problem, but they cannot handle the temporal dimension of the video case very well.

2.2.2 Recurrent Neural Network

Another model called Recurrent Neural Network (RNN) model is one of the most used for temporal analysis of data, it achieves temporal dynamics in a very ingenious way by embodying correlations between data points that are close in the sequence. The underlying dynamics of an input sequence are modelled by time evolving states which are learned by either a hidden layer or a memory cell, followed by mapping the states to outputs. RNN models have demonstrated competitive performances on speech recognition and text generation, however a significant limitation still exists: lack of ability

to backpropagate an error signal through a long-range temporal interval, which makes difficult learning long-term dynamics.

2.2.3 Long Short-Term Memory Network

To overcome this defect, Schmidhuber et al. [12] had proposed a specific recurrent architecture, namely Long Short-Term Memory (LSTM). In this networks, a special node called Constant Error Carousel (CEC) was designed to enhance the nonlinear mechanisms of hidden state, which allows constant error signal to propagate through time without modification, as by being updated or reset. They employed multiplicative gates to control the access to the CEC. Recently, LSTMs have been used widely and have shown tremendous potential in action recognition tasks. But LSTM still has a deficiency: it is unable to obtain the salient dynamic patterns.

2.2.4 3D Convolutional Neural Network

Extending CNNs, Baccouche et al. [3] proposed 3D convolutional neural network by taking space-time volume as input. This neural-based deep model was a two-step process. In the first part of the model, spatio-temporal features were learned automatically. The fact that they only took advantage of raw inputs made this model distinctive. A recurrent neural network model was then trained using these learned features to classify the entire sequence. The label of the entire sequence was learned by integrating several individual decisions, these individual decisions correspond to the inputs of the 3D-CNNs learning process for a small temporal neighborhood. The temporal evolution of the features were fully utilized rather than applying majority voting process on the individual decisions. In addition, to classify the action sequences, a LSTM cells with one hidden layer was introduced in a Recurrent Neural Network architecture, which significantly improves the performance.

Veeriah et,al. [43] developed a new family of LSTM model called differential RNN (dRNN) which could automatically learns the dynamic saliency of the actions performed. The authors introduced Derivative of State (DoS) that were sensitive to the spatio-

temporal structure of input sequence. Specifically, DoS allowed the gate units to learn the appropriate information, though quantifying the change in information gain caused by the salient motions between the consecutive frames. Followed by calculating different-orders of DoS, the dynamics of actions was modelled.

2.3 Dataset

The challenge and requirement for action recognition in different frames required the introduction of different human action video datasets.

At an early time, the study of action recognition was just getting started, and the analysis focused only on single-human and single-action. Therefore, the action video datasets created at that time were usually recorded in controlled conditions, where an individual actor performed one action before static, uncluttered backgrounds and captured by a fixed point of view. The two most known and used are Weizmann (2001&2005)[4] and KTH (2004) [39].

We aims at performing automated recognition of human actions in the static background. In order to test the effectiveness and efficiency of our approaches, we consider datasets captured by fixed camera, *e.g.* KTH dataset.

The Weizmann dataset [4] contains 10 different actions: walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack and skip. Example frames are shown in Figure 2.3. Each action is performed by 10 different actors leading to 100 clips in total. The viewpoints are static resulting in the static backgrounds in the dataset, and foreground silhouettes are also included. In addition to this dataset, there are two separate sets of sequences recorded for robustness evaluation. One set includes walking movement captured from different angles. The other one shows fronto-parallel walking actions with slight variations (carrying objects, different clothing, different styles).

The KTH action dataset [39] contains six classes of human actions: walking, jogging, running, boxing, hand waving and hand clapping, which are performed several times by 25 people in four different scenarios: outdoors, outdoors with scale variation,

Figure 2.3: Example frames from Weizmann action dataset.

outdoors with different clothes and indoors. There are 2391 sequences in total. The sequences have spatial resolution of 160 x 120 pixels and an average length of four seconds with 25 FPS frame rate. Typical clips are shown in Figure 2.4. For this dataset, there is only slight camera movement, apart from the zooming scenario. So the backgrounds are relatively static.

More recently, recognizing actions from realistic videos became of interest. The datasets are inclined to be sampled from existing video, such as sports broadcasts, TVs and Movies. Large variations appear in those videos, scale changes, dynamic backgrounds, illumination conditions, etc. These impact variations in the videos provide a more realistic environment but also much more challenging. The most extensively used datasets are: UCF sports action dataset[36] which contains 150 sequences of sport motions in bounding boxes of the human figure. Considerable variation are included for most action classes, in terms of action performance, human appearance, camera movement, viewpoint, illumination and background. Hollywood human action dataset[27] collects

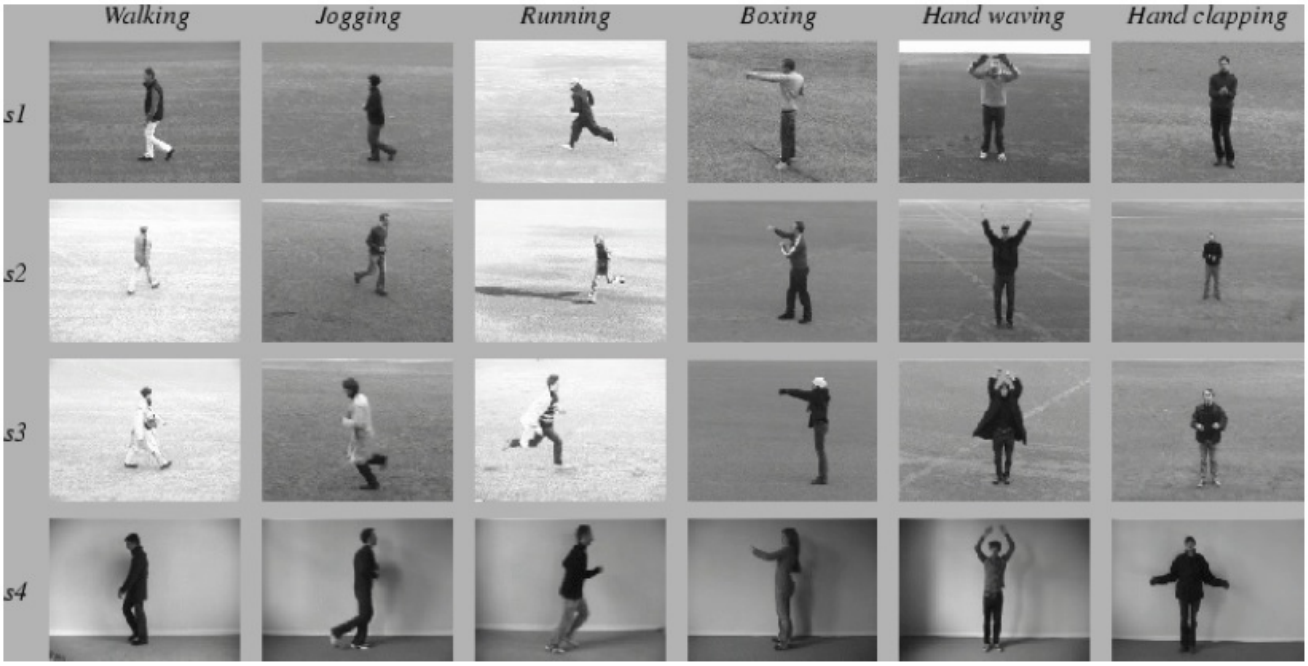

Figure 2.4: Example frames from KTH action dataset.

videos from Hollywood Movies 12 action classes, with a previous version[24], which contains 8 classes of that 12 ones. The second version is comparatively large, including 3669 high quality clips. Due to multiple persons, large intra-class variation, camera motion and dynamic backgrounds, it is a very challenging dataset. Figure 2.5 illustrates some examples frames of these datasets.

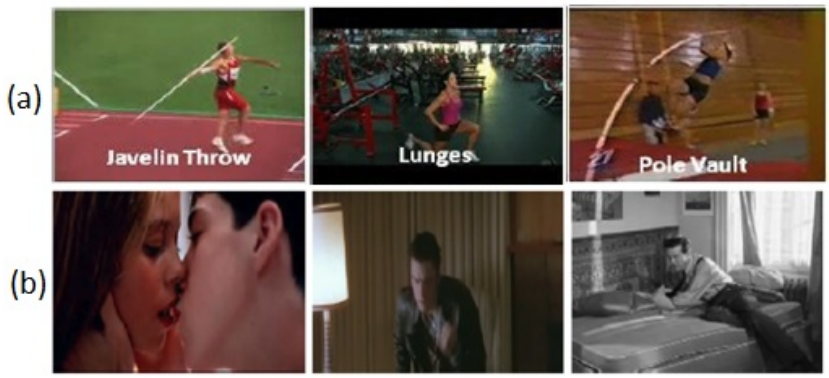

Figure 2.5: Example frames of (a) UCF action dataset and (b) Hollywood human action dataset.

Chapter 3

Video Feature Descriptor

A video feature descriptor encodes pixel representations in an image sequence to describe an action and maximize classification performance. A descriptor allows different classes of videos to be matched across each other, which means that it should ideally be invariant to changes, such as the appearance of a person, background, viewpoint, and action execution. With respect to precision, a descriptor must also be sufficiently rich to generate an exact representation and robust classification of the action (see Chapter 4); however, this should not come at the cost of efficiency. We propose a preprocessing step for improving efficiency by extracting ST regions of interest (ROIs), and we discuss the steps for building the descriptors. We also introduce some strategies used to explicitly represent the temporal dimension, as well as enhance the precision and efficiency of the

recognition.

3.1 Feature Descriptors

Dollar et al. [10] extracted dense features from normalized pixel values, image gradients, and windowed optical flows to build different descriptors. The results showed that concatenated gradients provided the best performance. Later, by comparing different types of descriptors based on local jets, optical-flow histograms, and gradient histograms, Laptev and Lindeberg [21] proved that optical-flow histogram descriptors and ST gradient descriptors outperformed others. Various descriptors have been developed by employing this fundamental background knowledge.

3.1.1 Local Feature Descriptors

The HOG descriptor is a popular 2D descriptor originally developed by Dalal and Triggs [8] for detecting human appearance and action. As illustrated in Figure 3.1, image gradients are first calculated by employing a [-1, 0, 1] filter along the x- and y-axis without smoothing. The image is then divided into a grid of blocks, each of which is split into a grid of cells. For each pixel in the corresponding cell, the orientation and magnitude are computed based on the intensity gradients, and the orientated histograms are voted with weighting according to gradient magnitudes.

Klaser [20], who considered a 3D gradient instead of a 2D gradient, proposed the HOG3D descriptor. The ST gradients are computed for each pixel of the video, and are then saved in three integral videos along the x, y, and t dimensions. A given 3D patch is split into a grid of Mc x Mc x Nc cells, with Mb x Mb x Nb sub-blocks per cell. The mean 3D gradient is then computed for each sub-block, and a regular polyhedron is used to quantize uniformly the orientation of the mean gradients into an n-bins histogram. The 3D HOG for the 3D patch is generated by concatenating gradient histograms of all the cells into a vector, and normalizing using the L2 norm. This way, shape and motion information are both included in the descriptor.

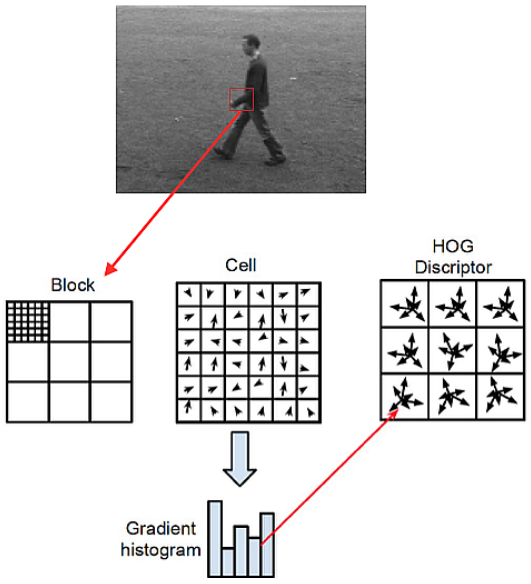

Figure 3.1: Block diagram of HOG method.

3.1.2 GBH Descriptors

In the recent studies, performance of human action recognition descriptors has reached a relatively high level of performance, particularly for the atomic actions captured under controlled settings. Most of the improvement has actually been with recognition precision, with less consideration on efficiency.

Shi et al. [41] proposed a new, highly efficient local ST descriptor, the GBH. It aims at improving performance while avoiding the high dimensions of common 3D gradient-based descriptors, as well as avoiding their relatively expensive quantization cost. In order to accomplish these goals, they abandoned the three-gradient components of in 3D SIFT and HOG3D descriptors; instead, they apply a compact HOG-like descriptor with two-gradient components.

To be specific, they first compute image gradients using simple 1-D [1, 0, 1] Sobel masks [44] on both the x and y directions for each frame in a video. A [1, 1] temporal filter is next employed over two consecutive gradient images. Thus, for each pixel, they compute:

$$I_{t,x} = \frac{\partial}{\partial t}\left(\frac{\partial I}{\partial x}\right), \qquad I_{t,y} = \frac{\partial}{\partial t}\left(\frac{\partial I}{\partial y}\right) \tag{3.1}$$

The gradient magnitude and orientation for each pixel are then defined as follows:

$$r(x,y) = \sqrt{I_{t,x}^2 + I_{t,y}^2}, \qquad \theta(x,y) = arctan\left(\frac{I_{t,y}}{I_{t,x}}\right) \tag{3.2}$$

The GBH descriptor still use the histogram of orientation-based method, voting with θ and γ with the HOG descriptors; however, instead of using 3D image gradients, time derivatives of image gradients are used in this process. Because the time derivatives of image gradients emphasize moving-edge boundaries, they called this descriptor the GBH; as a result, they encode both local static appearance and motion information in a relatively compact way.

The gradient boundaries is shown in Figure 3.2. It has less background noise due to the subtraction of two consecutive image gradients; in addition, the red bounding boxes in the Figure illustrate that gradient boundaries encode moving human shapes. Indeed, the moving speed of the human body parts is embodied by the various distances of the double edges.

Upon completing in-depth experimental evaluations, we noted that simple gradient subtraction worked well when the camera and background were largely static, but we also achieved good performance on realistic datasets.

Figure 3.2: Illustration of gradient boundaries.

3.2 Dense Sampling

A recent trend in action recognition consisting applying dense sampling to sparse interest point detectors to improve the performance of vision recognition. Dense sampling applies the sampling process for every pixel in each spatial temporal scale, which capture considerably more information than interest point detectors alone. This way, dense sampling of regular space-time grids outperforms state-of-the-art interest point detectors for action recognition. Wang et al. proved proof of this in [48], and similar results are also be observed in [47]. Although it has achieved relatively significant results, the main drawback of this approach is its intractability for large video datasets still exists.

There are five parameters for determining a feature point. The (x, y, t) parameters determine the center of a 3D video patch, and (σ, τ) is the multi-scale factors that govern a sampled patch size. Another factor that controls the sampling density is the overlapping rate of sampling patches.

3.3 PCA

Principal component analysis is a statistical method that converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Once these patterns in the data is found, the data is able to be compressed. The popular application of this technique is to reduce the number of dimensions in image compression without much loss of information.

In [18], the authors had proved that the PCA-based local descriptors were more compact than the standard representation with more distinctive and robust image deformations. They ran different types of experiments to explore the difference between the standard SIFT representation and PCA-SIFT. The descriptors' robustness was examined under the condition of noise, changes in illumination and the application of image transformations. They also evaluated descriptors' generation on real images taken from

varying viewpoints and an image retrieval application. Compared to the standard representation, PCA-based local descriptor is both more distinctive and more compact leading to significant improvements in matching accuracy (and speed) for both controlled and real-world conditions.

3.4 Proposed method

Our framework can be summarized in the following steps:

- HOG people detection is applied to detect active people, and 3D ST patches are generated according to the detecting results.
- GBH descriptors are then built for each 3D ST patch.
- PCA is finally employed to reduce the dimensions if necessary.

We will give more details in the sections that follow.

3.4.1 The 3D ST Patch

The process of generating a 3D ST patch is shown in Figure 3.3. The green box displays an example of a 3D ST patch containing a few consecutive frames of active people. More specifically, each video is first divided into a set of sub-sequences with setting frames. We next apply HOG people detection to the first frame of those sub-sequences in order to detect active people. The region of the people detected by HOG in the first frame is then regarded as the ROI. That ROI is also used for the rest of the consecutive frames in each sub-sequence. The ROIs with active people are isolated and then used as the 3D ST patch.

In order to improve performance of HOG people detection, a strategy called non-maximum suppression (NMS) [37] is applied. HOG uses sliding windows approach to detect the presence of objects, which typically results in multiple windows with high scores close to the correct location of objects. This is due to the smoothness of the response function, the generalization ability of object detectors, and the visual correlation

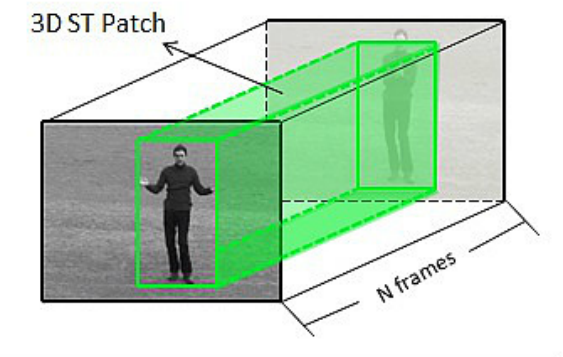

Figure 3.3: An example of a 3D ST patch.

of nearby windows. This relatively dense output influences the understanding of an image, because the number of window hypotheses in this step is not correlated to the real number of objects in the image. Because of this, we employed NMS to retain only one window per group according to the precise local maximum of the response function with the purpose of obtaining only one detection per object, and enhancing the detection precision.

We try to avoid using three-gradient components such as HOG3D descriptors, which would lead to high dimensionality and relatively expensive quantization cost. We only apply HOG to the first frame of each small sequence. Since each sequence contained only a few frames, we can generate 3D ST patches to build rich descriptors.

3.4.2 Proposed GBH Descriptor

The GBH descriptor is built on simple ST gradients. It exhibits good performances in large realistic datasets, and provides a better representation of local structure and motion.

The process can be summarized as follows, and is shown in Figure 3.4.

- The image gradients are computed using simple 1-D [-1,0,1] Sobel masks for each pixel over the input sequence. The integral video strategy is applied.
- A [-1, 1] temporal filter is employed over two consecutive gradient images to obtain the frame differences.

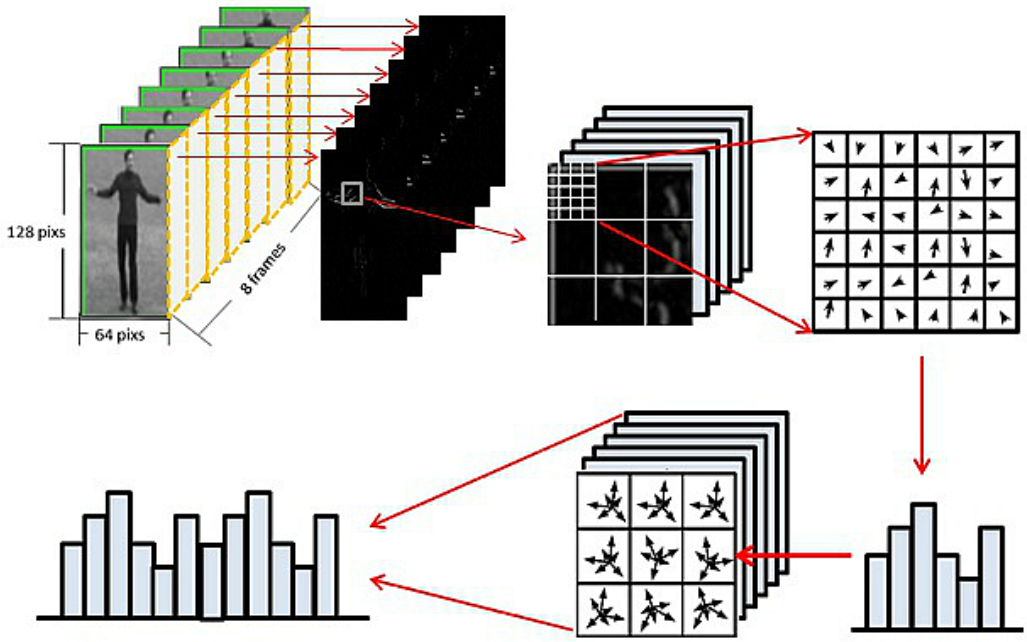

Figure 3.4: GBH descriptor computation.

- Dense sampling is assembled for each gradient patch at different locations, and multiple spatial and temporal scales are investigated.
- Each sample of the sequence is divided into a grid of M_b x M_b x N_b blocks, where each block was again split into a grid of M_c x M_c x N_c cells.
- The image gradients of each cell are quantized using a regular 2D polygon, and each side of the polygon corresponded to a histogram bin. The histogram for this block is obtained by summing the quantized gradients of all its cells.
- The histogram of every block is cascaded to create the histogram for the sample.
- The GBH descriptor for the input 3D ST patch is built by concatenating the gradient histograms of all samples.
- The histogram is normalized by L_2.

3.4.2.1 Gradient Computation over Integral 3D ST Patches

The patches we previously obtained need to be sampled with different spatial and temporal scales at various locations and times. According to [20], one possible strategy is to apply ST "pyramids" to improve computational efficiency, such as precomputed gradients on different temporal and spatial scales [25]. With this method, the input sequence needs to be rescaled and stored in each ST scale. To be specific, given N scale steps in total with a spatial and a temporal scaling factor of σ_{xy}, σ_t, the total memory needed would be increased by a factor of:

$$z = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} {\sigma_{xy}}^{-2i} \sigma_t^{-j} \tag{3.3}$$

In [20], integral video strategy was proposed as being not only memory-efficient, but also a flexible alternative. The authors extended the concept of the integral image proposed by Viola and Jones [45] for efficient computation of Haar features. The integral video iv in this strategy can be described as:

$$iv(x, y, t) = \sum_{t'<=t} \sum_{y'<=y} \sum_{x'<=x} i(x', y', t'), \tag{3.4}$$

where $\mathrm{i}(\mathrm{x}', y', t')$ is the pixel value at the location (x', y', t') of the input image sequence. Therefore an integral video sequence at spatial the location *(x, y)* and time t is defined as the sum of all the pixel values at locations less than or equal to *(x, y, t)*. Given the 3D rectangular volume $\mathrm{c} = (\mathrm{x}_0, y_0, t_0, w, h, l)$ with width w, height h, and length l at location of (x_0, y_0, t_0), its sum can be computed as:

$$\begin{aligned} S_0 = {} & iv(x_0 + w, y_0 + h; t_0 + l) - iv(x_0, y_0 + h, t_0 + l) - (x_0 + w, y_0, t_0 + l) \\ & + \mathrm{iv}(\mathrm{x}_0, y_0, t_0 + l) - iv(x_0 + w, y_0 + h, t0) + iv(x_0 + w, y_0, t_0) \\ & + \mathrm{iv}(\mathrm{x}_0, y_0 + h, t_0) - iv(x_0, y_0, t_0). \end{aligned} \tag{3.5}$$

Therefore, given a 3D ST patch, we first apply 1-D [-1, 0, 1] Sobel masks to every frame, and capture the image gradients for every pixel. To build the time derivatives of the image gradient, a [-1, 1] temporal filter is then used for every two successive frames.

3.4.2.2 Dense Sampling

After obtaining the gradient patches, we employ the dense sampling at different scales and locations to extract ST samples, as described in [20]. A sampled gradient patch $s = (x_s, y_s, t_s, \sigma_s, \tau_s)$ is located at (x, y, t), with characteristic spatial and temporal scales given by σ, and τ, respectively. The features around position s are computed for a local ST region with the size of width w_s , height h_s and length l_s given by

$$w_s = h_s = \sigma_0 \sigma_s \qquad and \qquad l_s = \tau_0 \tau_s \tag{3.6}$$

where σ_0 and τ_0 are the initial spatial and temporal scales, respectively, and σ_s and τ_s are the spatial and temporal step factors for successive scales. We define the initial values of τ_0 and σ_0, and set scale steps as $\sigma_s = \tau_s = (\sqrt{2})^i$, given $i = 0, 1, 2, ...,K$ as the ith step of the scales.

3.4.2.3 Orientation Quantization

To construct the GBH descriptor, an n-bin histogram of gradient orientations is built. The objective of the GBH descriptor is to avoid using three-gradient components like HOG3D descriptors, which would lead to high dimensionality and relatively expensive quantization costs, even though it would capture both local static appearance and motion information at the same time; therefore, we adopt the compact HOG-like descriptor with two-gradient components.

The histogram used here is seen as an approximation of a circle, with a regular 9-sided polygon. Each side of the polygon, at that point, corresponds to a histogram bin with a range of either $0 - 180°$ or $0 - 360°$. Each gradient of the image's time derivatives calculates a weighted vote for an orientation histogram channel based on the orientation of the gradient element centred on it, and the votes are accumulated into orientation bins over local spatial regions. To reduce aliasing, votes are interpolated bilinearly between the neighbouring bin centres in both orientation and position. The vote is the gradient magnitude at the pixel.

3.5 Conclusion

In this chapter, we presented a novel method for human action recognition in video with static backgrounds. We proposed removing the background information before analyzing the action, followed by the introduction of a new local descriptor based on an ST gradient for capturing both the local static appearance and motion information. This descriptor can be efficiently computed.

Chapter 4

Representation and Classification for Action Recognition

In the previous chapter, we have presented our GBH descriptors for representing 3D ST patch. We will use this descriptor to achieve action recognition. Figure 4.1 shows the framework of the proposed method. In this chapter, the steps inside red bounding box will be introduced in detail.

One representation method is BoF, which has become more and more popular and has achieved excellent levels of performance. BoF uses statistical information of local features to represent videos, and it has good tolerance for variant conditions in video, such as illumination, occlusion, deformation, and multiple motions.

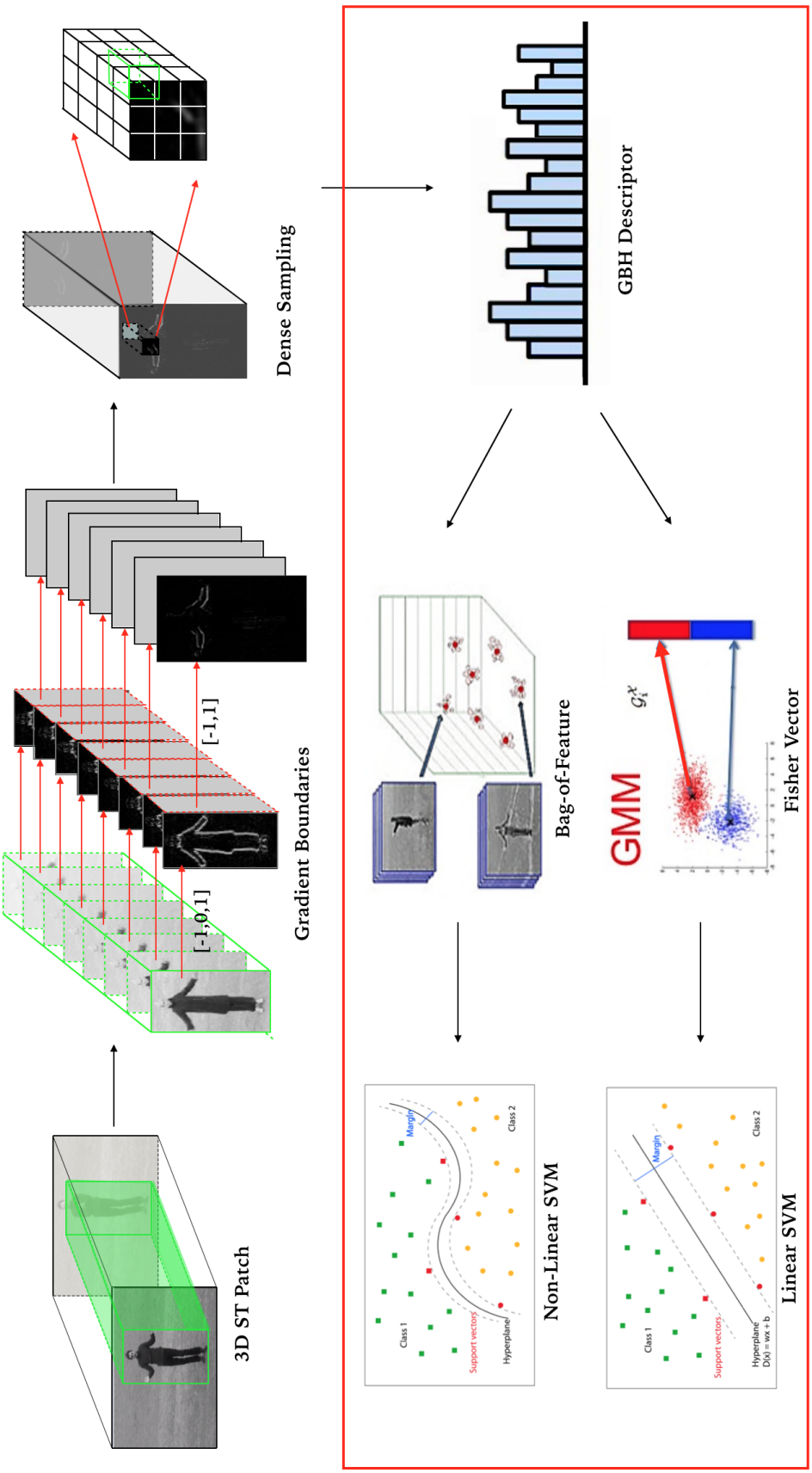

Figure 4.1: Illustration of the proposed method.

As an alternative patch encoding strategy, the FV has recently been attracting more attention. In a recent study that evaluated feature-pooling techniques for object recognition, the FV proved to be the most effective encoding technique [32]. It extends the BoF by encoding high-order statistics (first and, optionally, second order) between the descriptors and a Gaussian Mixture Model (GMM). This means that, not only the occurrences of code words are coded, but also the additional information about the distribution of the descriptors; additionally, it is also efficient for computation because it works very well with efficient linear classifiers, and it can be compressed with a minimal loss of precision using product quantization.

Efficient human action recognition system is also important; therefore, a strategy called PCA is investigated for reducing the high feature dimensionality. The results are substantial benefits in computation efficiency.

4.1 The BoF Approach

The features extracted from a video must be represented in a way that supports classification. To achieve this, the video often ends up being represented as a single vector. During the past decade, the BoF approach has attracted great attention from the computer-vision community because of its special characteristics. First, explicitly modeled objects and their relationships to the complex content of images are hard to deal with when it comes to image content analysis. Utilizing small, characteristic image regions in BoF representation avoids these tasks. The second important characteristic of this approach is its adaptiveness to the particular image collection to be processed. It allows for the identification of visual patterns that are relevant to the whole image collection, which means that the patterns that are used in a single image representation come from the analysis of patterns in the complete collection; moreover, the robustness to occlusion, affine transformations, and computational efficiency are also important advantages of the BoF approach.

BoF proceeds as follows: A K-means clustering algorithm is applied to all the feature vectors (descriptors) to define a vocabulary. The histogram is generated by quantizing

the feature vectors into their closest bin according to their Euclidian distance; finally, normalization is applied to make all feature vectors comparable.

4.1.1 Visual Vocabulary Construction

The visual vocabulary, or codebook, is built using either a clustering or a vector quantization algorithm, which is intended to identify a set of visual patterns that reflect the contents of the image collection. An important decision in construction is the selection of the size of the codebook. It was reported in [7] that performance improved steadily as the codebook grew. In [30], the authors ran similar experiments to study the relationship with the number of patches sampled in the test image, and they obtained substantial gains in performance as the codebook size was increased. We evaluated different codebook sizes to analyze the impact of this parameter in the classification.

4.1.2 Image Representation

The final representation is calculated by counting the occurrence of each codebook center in the set of local features. This representation ignores the geometric relationships of local features. Normalized term frequencies are also calculated by normalizing according to the number of features extracted from the data.

4.2 The FV Encoding

In a recent evaluation study [32], the FV, as an extension of BoF, was found to be the most effective encoding technique. It encodes high-order statistics (first order and, optionally, second order) between the descriptors and a GMM; therefore, both the information for code word occurrences and the distribution of the descriptors can be encoded at the same time.

4.2.1 The Fisher Kernel

The principle of using the Fisher kernel framework to build Fisher Vector is explained in [16]. The authors assumed that $X = x_t, t = 1...T$ was the set of T local descriptors extracted from an image, and the generation process of X could be modeled by an image-independent probability density function u_λ with parameter λ[2]. Jaakkola and Haussler[14] proposed describing X by the vector:

$$G_\lambda^X = \frac{1}{T}\nabla_\lambda log u_\lambda(X). \tag{4.1}$$

Based on [14], given the gradient of the log-likelihood, which describes the contribution of the parameters to the generation process, the "Fisher kernel" is the natural kernel for these gradients:

$$K(X,Y) = G_\lambda^X F_\lambda^{-1} G_\lambda^Y \tag{4.2}$$

where F_λ is the Fisher information matrix of u_λ. As F_λ^{-1} is positive semi-definite, it has a Cholesky decomposition $F_\lambda^{-1} = L'_{lambda} L_\lambda$. The FK $K(X,Y)$ can therefore be rewritten explicitly as a dot-product:

$$K(X,Y) = \mathcal{G}_\lambda^{X'} \mathcal{G}_\lambda^Y \tag{4.3}$$

where

$$\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X \tag{4.4}$$

$\mathcal{G}_\lambda^X$ is regarded as the Fisher Vector (FV) of X.

4.2.2 The FV representation

In computer vision literature, such as [31], a GMM demonstrating the generation process of local descriptors in any image is referred to as a universal (probabilistic) visual vocabulary. We follow the authors of [31] and chose u_λ to be the GMM:

$$u_\lambda(x) = \sum_{i=1}^{K} \pi_i u_i(x) \tag{4.5}$$

The parameters of the K-component GMM are donated by $\lambda = \pi_i, \mu_i, \sigma_i, i = 1...K$, where π_i, μ_i and σ_i are, respectively, the mixture weight, mean vector, and variance matrix (assumed diagonal) of the Gaussian u_i. Here, the gradient in terms of the mean and the diagonal closed-form approximation of the Fisher information matrix are used, in which case the normalization of the gradient by $L_\lambda = F_\lambda^{-1/2}$ is simply a whitening of the dimensions. The soft assignment of descriptor x_t to the i^{th} Gaussian can be described as follows:

$$q_{ti} = \frac{\pi_i u_i(x_t)}{\sum_{j=1}^{K} \pi_j u_j(x_t)} \tag{4.6}$$

After normalization with the inverse Fisher information matrix (which rendered the FV invariant to the parameterization), the gradients for the mean and variance of the i-th Gaussian are given by:

$$\mathcal{G}_{\mu_i}^X = \frac{\sum_{t=1}^{T} q_{ti}[x_t - \mu_i]}{\sqrt{\sigma_i \pi_i}}, \tag{4.7}$$

$$\mathcal{G}_{\sigma_i}^X = \frac{\sum_{t=1}^{T} q_{ti}[(x_t - \mu_i)^2 - \sigma_i^2]}{\sqrt{2\sigma_i^2 \pi_i}}. \tag{4.8}$$

4.2.3 FV Normalization

Two normalization steps must be applied before building the final Fisher Vector, as proposed in [33]. The first is power normalization, which applies the operator of $f(z) = sign(z)|z|^\alpha$ with $0 < \alpha \leq 1$ independently on each dimension of the FV. After this step, the influences of bursts of visual elements are reduced, and the dependence between the variance and the mean are eliminated, since the power normalization can be interpreted as a variance stabilizing transformation. This way, the quality of the representation is consistently improved. The power normalized FV is, finally, $L2$-normalized, which amounts to using the cosine similarity between FVs. On one hand, $L2$-normalization is justified as a way of compensating for the fact that different images contain different amounts of background information. On the other hand, $L2$-normalization is valid for

any high-dimensional vector. These two complementary interpretations explain why such normalization can lead to improved results.

4.3 Support Vector Machine (SVM)

Support Vector Machine is a discriminative classifier defined by a hyperplane that separates the classes better than other classifiers. Given labeled training data, the algorithm outputs an optimal hyperplane that categorizes new examples.

4.3.1 Linear separable classes

In a simple case, it is possible to achieve complete separation between two classes using a linear function. When the two-dimensional space extends to n-dimensional space, a hyperplane is used as the linear function. Usually, the hyperplane in a design is located in the middle of two point clouds that are the maximum distance away. The position of the hyperplane is influenced by points, known as support vectors, which are closest to the empty space between the classes. The distance between the hyperplane and the support vectors is the margin, so the desired hyperplane is defined as the hyperplane that maximizes the margin.

4.3.2 Non-separable points

Practical cases tend to be more complex than the ones previously discussed, in which the two classes are completely separated by a hyperplane. Points are often positioned in a way that makes it impossible to separate them; therefore, the positive-slack variable is introduced. This slack allows the points to be located on the "wrong" side of the hyperplane, and it improves the accuracy of the classifier in these situations

4.3.3 Extension to non-linear

A linear decision function is not always enough to solve the problems resulting from the appearance of non-linear data. These problems can also be resolved by applying a

mapping implicitly in the optimization. A function k is created, and can be described as:

$$k(x, y) = \phi(x)\phi(y) = \phi(x)^T\phi(y) \tag{4.9}$$

The function k is called a kernel-function, which maintains the complexity of the original space and achieves higher classification accuracy. Table 4.1 shows some popular choices of SVM kernels.

Kernel	Expression
Linear	$k(x, y) = x^T y + c$
Polynomial	$k(x, y) = (ax^T y + c)^d$
Radial basis function(RBF)	$k(x, y) = exp(-\frac{\|\|x_i - y_i\|\|^2}{2\sigma^2})$
Intersection	$k(x, y) = \sum_{i=1}^{n} min(x_i, y_i)$
$\chi - squared$	$k(x, y) = 1 - \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$

Table 4.1: Different examples of kernel functions.

4.3.4 Multi-class approach

Originally, SVMs were developed to perform binary classification; however, binary classification applications are very limited when classification problems involve more than two classes. With respect to the general case of n classes, two common approaches can be applied: one-vs-all and one-vs-one.

For the one-vs-all approach, n binary SVM classifiers may be created where each classifier is trained to distinguish one class from the remaining $n1$ classes. For example, a class one binary classifier is designed to discriminate between class one data vectors and the data vectors of the remaining classes. During the testing or application phase, data vectors are classified by determining the margin from the linear separating hyperplane. The final output is the class that corresponds to the SVM with the largest margin.

For the one-vs-one approach, SVM classifiers for all possible pairs of classes are created, resulting in $n(n1)/2$ classifiers. Each point casts a vote for one of the classes in

each of the classifiers, and the class with the highest number of votes is finally selected. In case of a tie, a tie-breaking strategy may be adopted. A common tie-breaking strategy is to select randomly one of the class labels that are tied.

4.4 Action Recognition Framework

4.4.1 BoF Representation

We use a standard BoF method to represent a video sequence.

- Visual vocabulary is built using all the GBH descriptors extracted from the training videos. All the descriptors are sent to the K-means system [2] and clustered into k centers. The word vocabulary of k visual words is then created using these centers.
- A set of GBH descriptors from a video sequence are assigned to the closest word from the vocabulary using the Euclidean distance, and the histogram of visual word occurrences is generated to represent the video.

4.4.2 FV Encoding

In addition to using the BoF, we also evaluate the FV for encoding features. Since the FV signature with K words can generate an increase of 2KD on the dimension of a D dimension descriptor; therefore, PCA was applied to the computed GBH features. The VLFeat library [42] for the FV encoding is used. We also applied the signed square root and L2 normalization, as introduced in [32].

4.4.3 SVM Classification

The SVM classification is applied in our work. For classification with the FV, we use the linear SVM with the fixed parameter $C = 16.25$:

$$k(x, y) = x^T y + c \tag{4.10}$$

For the feature represented by the BoF approach, we employ the SVM with the Radial Basis Function (RBF) kernel:

$$k(x, y) = exp(-\frac{||x_i - y_j||^2}{2\sigma^2}) \tag{4.11}$$

The LibSVM [6] is utilized for the SVM classification, which takes advantage of one-vs-all for multi-class approach. The vector x_i is the occurrence of the visual word histogram, as y_i is a support vector. A cross-validation procedure is also applied to the training data to determine the best parameters for the SVM.

4.5 Conclusion

We discussed two methods for representing video features: the BoF approach and FV encoding. The BoF is a relatively simple approach to build a rich representation. It however requires non-linear SVMs to achieve good performance. On the other hand, the FV produces a more complex representation, but works well with simple linear SVMs. The next chapter will present our experimental experimentations that use and compare these two approaches.

Chapter 5

Experimental Evaluation

In this chapter, the experimental results are evaluated by applying different parameter settings for the proposed method.

We first generate 3D ST patches which are composed of several frames recording active people, and compute the time derivatives of the image gradient for the patches. Dense sampling is then employed over these gradient patched, with multiple spatial and temporal scales. The GBH descriptor is now built on each sample, and all the descriptors are concatenated to describe the corresponding patch. Features is able to extracted by encoding these descriptors, to represent various actions. We produced both the BoF and FV during the encoding process. Finally, different types of SVM classifiers are applied. The performance is evaluated by setting various parameters for dense samplings and

visual words of representation. Additionally, PCA is used to build GBH descriptors during this process.

The experiment is performed on KTH dataset. According to the directions given in the dataset [39], we divide the videos into testing set (9 subjects: 2, 3, 5, 6, 7, 8, 9, 10, and 22) and training set (the remaining 16 subjects). The 600 video files are not separated into sequences, which means that one video file contains the same subject performing an action multiple times. Additionally, the computational time is estimated on an Intel i5-2500 PC with 16GB memory. The prototype is implemented in C++ with OpenCV library.

5.1 GBH Descriptor

5.1.1 3D Spatio-Temporal Patches

As illustrated in Figure 5.1, each video is divided into sub-sequences of 8 frames, and HOG people detection is applied to the first frame of every sub-sequence. A 3D ST patch for each sequence is extracted based on the initial detection. To precisely detect people, a relatively small sliding window is more applicable for HOG detection, which results in better performance but denser output. NMS is then used to select the best output per group. The effect of NMS is shown in Figure 5.2. The patches are resized into 64 x 128 x 8, corresponding to 8 successive frames of 64 x 128 pixels each. 16,868 patches are obtained by applying this step over 187,253 frames of the training set.

5.1.2 ST Gradient Computation over the Patches

Image gradients of video patches are first computed. For each frame in the patches, 1-D [-1,0,1] Sobel mask is assembled on both x and y directions, following by using [-1, 1] temporal filter over two consecutive gradient images. Figure 5.3 (a), (b) and (c) illustrate image gradient boundaries for handclapping, walking and running, respectively. In all these three video sequences, background is removed due to the subtraction of two consecutive image gradients. The other observation is that gradient boundaries embody

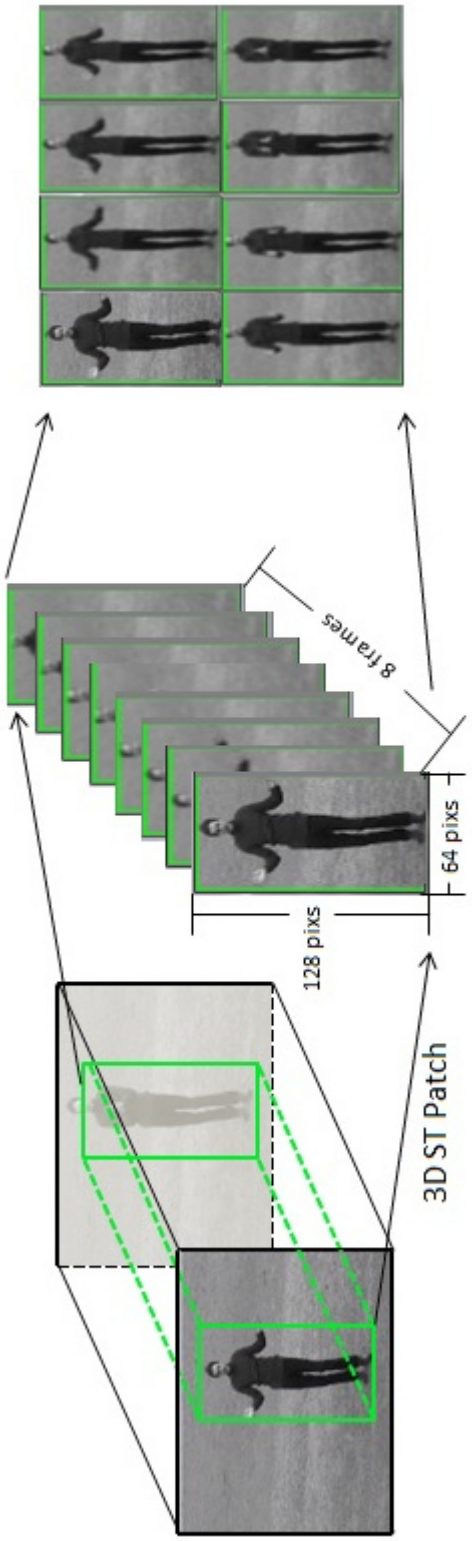

Figure 5.1: Example of the extracted 3D spatio-temporal patch.

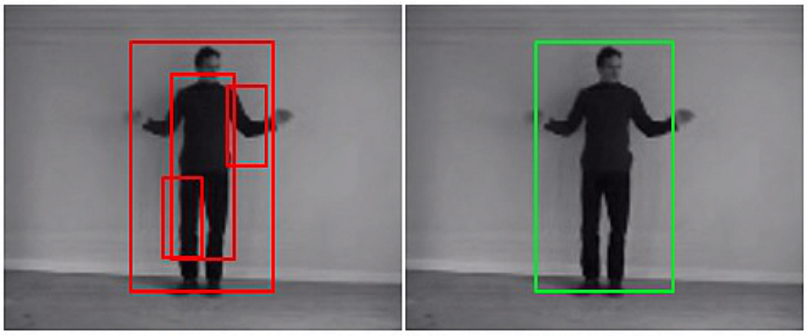

Figure 5.2: Example of the NMS effect.

the moving human shapes. As seen in the handclapping figure, the edges of the arms are emphasized since arms are the most active human body parts in this action. Comparing walking and running, more double edges can be seen in the latter, as the double edges are proportional to the moving speed of the human body parts.

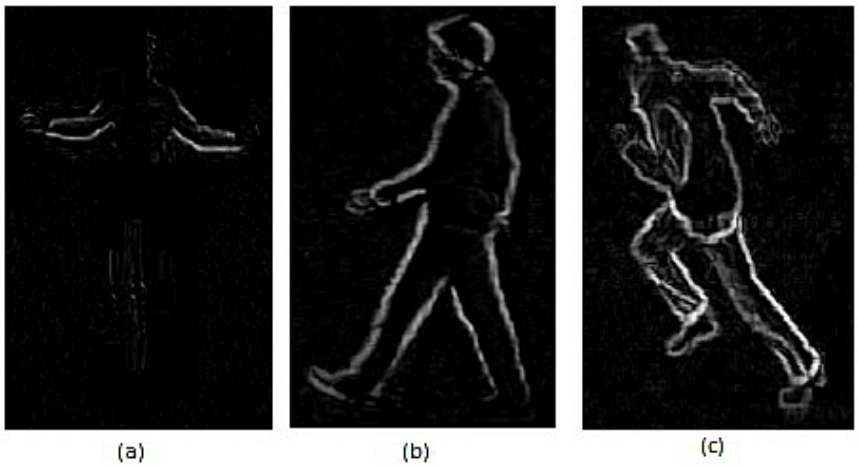

Figure 5.3: Example of image gradient boundaries for (a) handclapping (b) walking and (c) running in KTH dataset.

5.1.3 Dense Sampling

Each gradient patch output from the filters is then sent to the sampling process. We closely follow the parameters established by Wang et al. [48] for dense sampling. Scale factor of $\sqrt{2}$ is defined to obtain multi-scale samples. Spatial scales and temporal scales

are set into 8 and 2, respectively, as spatial scale is considered to be more important than time scale. Overlapping rate is 50%, which is used for spatio-temporal dense samples. The last parameters are the minimal value of spatial size and temporal length. They need to be minimal enough to get adequate samples, while keeping the complexity low. By investigating the performance as a function of different values, 16 and 4 are the most appropriate ones as the minimum spatial size and temporal length.

Following the parameter settings above, 582 samples are settled for each 64 x 128 x 8 gradient patch.

5.1.4 Histogram Computation

In this step, gradient boundary histogram for each sample in ST patches is generated by using the gradient values computed above. To do so we first divide each patch into a grid of 2 x 2 x 2 cells, with 2 x 2 x 1 sub-block divisions. For each pixel in the sub-blocks, both orientation and magnitude are computed based on the intensity gradients. Gradients vote for the 9 bins orientated histogram, by using a vote size corresponding to the gradient magnitude. The histograms of all the cells are then concatenated into one 2 x 2 x 2 x 9 = 72 dimensional histogram to represent the sample, following by cascading the histograms of all samples into GBH descriptor. Therefore, the final result is that using a 72 x 582 = 41,904 dimensional GBH descriptor to represent each patch.

Computational complexity of GBH descriptors is further analysed. The average number of frames per second is recorded when densely sampling and generating the descriptor. The result presents a comparatively high processing speed (47 frames per second) for the dense sampling case, which confirms the high efficiency of GBH descriptor, and its potential for action recognition and real-time applications.

5.2 Feature Representation

As the result above, 16,868 descriptors are obtained to encode features. We apply the Bag-of-Feature (BoF) and Fisher Vector (FV) techniques on those descriptors.

5.2.1 The BoF Approach

We follow the standard BoF process, and K-Means is employed for clustering.

Different codebook values are investigated to evaluate performance and efficiency. The codebook size is set at 100, 200 and 300. As illustrated in Table 5.1, there is a clear benefit when using 300 visual words. This result comes no surprise, since more codewords yield less error from bag-of-feature vector quantization. Figure 5.4 shows the confusion matrix of the best result, which is obtained from non-linear SVM classification on KTH dataset. The precision of the boxing, handclapping and handwaving classes is perfect, while the other three have a lower precision. A possible explanation for this is that the first three actions have adequate variation, which is more prone to build precise features. In contrast, the movement of the last three actions is similar only with differences in speed, which may result in similar patches being assigned to different visual words in different clips.

Number of codewords	100 words	200 words	300 words
Precision	87.5%	88.9%	90.6%

Table 5.1: Precision on KTH dataset with 100, 200, 300 codewords.

PCA is intended to reduce the computational cost of brute-force matching by reducing the feature dimensionality. In order to find the best configuration, we analyze the performance under different levels of PCA. The dimension of the descriptor is reduced by 1/8, 1/32, 1/64, respectively, and the results are shown in Table 5.2. In terms of the first three columns, the results are as expected: higher dimensional feature vector produce better precision, due to the high fidelity of the representation with high dimensional features. However, we observe an unexpected outcome when applying 1/64 PCA, which obtains a highest precision. Our hypothesis for this is that the first several components of the PCA subspace are sufficient for encoding the variations in the gradient patch, while the later components represent details in the gradient image that are not useful. In addition, using fewer components requires less storage, and results in faster matching. To achieve a trade-off between running time and recognition precision, the

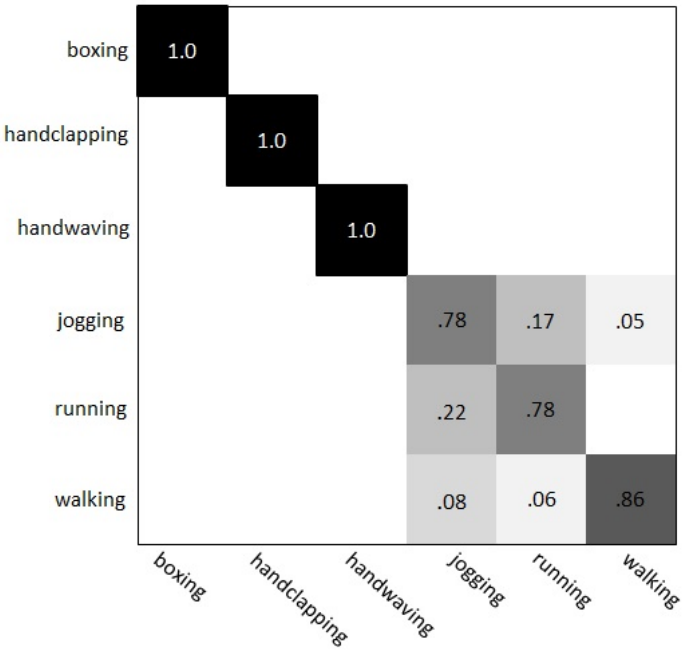

Figure 5.4: Confusion matrix of the best SVM classification with Bag-of-Feature on KTH dataset. Avg. precision: 90.6%

feature dimensions reduced by 1/64 through PCA leads to the best performance, which preserves the high precision and accelerates the matching process up to 8x.

PCA	1/8 PCA	1/16 PCA	1/32 PCA	1/64 PCA
Precision with non-linear SVN	88.7%	87.7%	86.2%	89.4%
Speed (fps)	32.08	86.49	156.61	263.65

Table 5.2: Average computation speed and precision of BoF on KTH dataset with PCA. The speed is measured in frames per second.

5.2.2 The FV Encoding

The FV method [32] is then evaluated. GMM is first applied for clustering, and the number of visual words k is set into 20 to estimate the GMM. The FV are then computed based on the result of GMM. Signed square-rooting is also applied, following by L2 normalization.

PCA is also used to reduce dimensions of descriptors, since the resulting high dimensionality is the main challenge for FV, which increases the expense in the classification stage even with linear SVM. Let D_0 be the feature dimension of GBH descriptor, we reduce it into $d^{'} = rD_0$, where r is the rate of reduction. Each video is then represented by a $2d^{'}K = 2rD_0K$ dimensional Fisher vector. In the result above, the dimension of GBH descriptor for each patch is 41,904.

Table 5.3 shows the results by applying different rates. In all experiments, the precision is relatively high even with linear SVM, and best results are obtained when dimensions are reduced into 1/16. Figure 5.5 illustrates the confusion matrix for the case of linear SVM classification on KTH dataset. FV not only preserves the high precision for the first three classes, but also achieves an important improvement for the last three. Meanwhile, FV has a very good computational speed performance, operating nearly twice as fast as BoF.

Rate of PAC	1/16 PCA	1/32 PCA	1/64 PCA
Precision With Linear SVM	92.0%	89.0%	90.8%
Speed (fps)	161.12	284.20	380.06

Table 5.3: Average computation speed and precision of Fisher Vector on KTH dataset with PCA. The speed is measured in frames per second.

5.2.3 Comparison of the BoF and FV

The common idea of these two methods is mapping the feature point into a fixed dimension vector, from which the distribution in the original feature space can be reconstructed approximately. However, the methods have several differences.

In theory, BoF uses K-Means to generate a hard quantization of feature space, where all clusters have similar importance, and their centroids are the only influential factor. In contrast, Fisher Vector employs GMM to build the descriptive vectors which are the gradient of the sample's likelihood, with respect to the parameters of this distribution and scaled by the inverse square root of the Fisher information matrix. This way, the specific learned distribution is designed to better fit the observed data in parameter

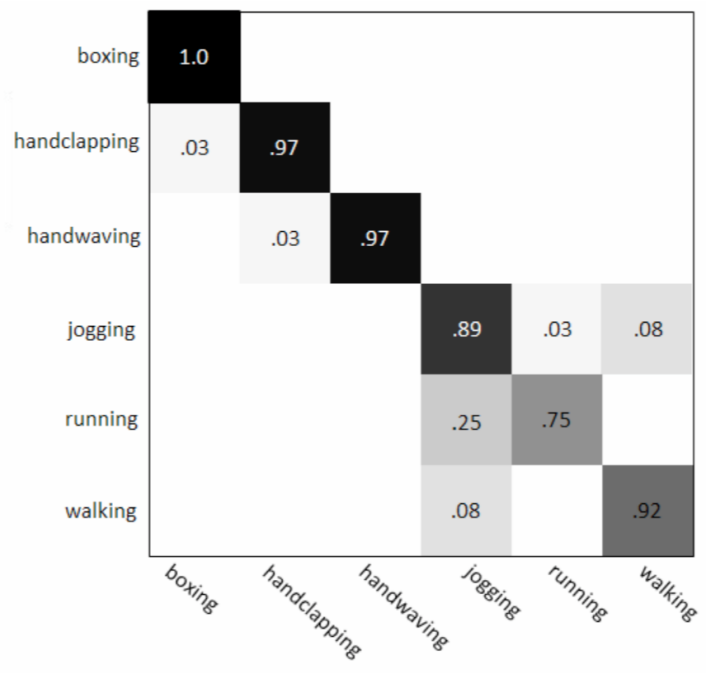

Figure 5.5: Confusion matrix of the best linear SVM classification with Fisher Vector on KTH dataset. Avg. precision: 92%

space. In addition, BoF approach only considers the number of assigned descriptors for each quantization cell, where FV takes advantage of both their mean and variance along each dimension. Hence, FV stores more information per cell, and then requires smaller number of quantization cells.

In practice, the benefits of FV are even more obvious. Under the same situation (1/64 PCA-based feature with linear SVM), FV gains 4% more precision (90.8% *vs* 86.3%) and higher computational speed (380.06 *vs* 263.65 frames per second) than BoF. Additionally, in contrast to K=300 codewords of BoF, FV only use K=20 visual words to obtain even better precision (90.6% *vs* 93.8%, without PCA), and a substantial increase in speed. In conclusion, FV has much better performance, not only in precision but also in efficiency.

5.3 SVM classification and Overall Result

In all experiments we train SVM classifiers, and set the regularization parameter by cross-validation. For multi-class SVM, one-against-all approach is used to select the class

with the highest score, which shows significantly better performance than one-against-one in action recognition [15]. Here, both linear and non-linear SVM classifications are estimated by applying BoF, and 300 codebooks and features reduced by 1/64 are used here. We could see a significant gap of speed between the linear and non-linear SVM, and only 3% penalty on precision.

BoF and FV are combined with both linear and non-linear SVM, and Table 5.4 shows the results of precision and efficiency. FV performs very well even with linear SVM. The extension of dimensions in this method procduces the complete separation of the classes, linear function can then be used in this situation. While BoF can only work with non-linear SVM since the data here is non linear, and kernel functions need to be applied to maintain the complexity of the space and achieve higher precision.

	Linear SVM		Non-Linear SVM	
	Precision	Speed	Precision	Speed
BoF	86.3%	56.9	89.4%	5.4
FV	90.8%	57.6	92.6%	22.9

Table 5.4: Comparison of the overall result between BoF and FV on average computation speed and precision. The speed is measured in frames per second.

At last, we compare the HOG3D with our method. From Table 5.5, we prove that the efficiency of our approach is very competitive, and we can use it to achieve the real-time recognition. Meanwhile, we manage to maintain the high precision.

	Precision	Speed(fps)
HOG3D	85.3%	0.8
Our Method	90.8%	57.6

Table 5.5: Comparison of the result between HOG3D and our method on average computation speed and precision. The speed is measured in frames per second.

5.4 Discussion

From the results presented in this chapter, we conclude that:

- The GBH descriptor shows good performance by improving efficiency and providing rich motion information. This efficient descriptor can be widely used in many different applications in video analysis.

- The FV consistently shows better performance than the BoF approach. In comparison with the BoF, the FV representation obtains higher precision with much less visual words and uses the simple linear SVM. The performance are even better by reducing the high dimension into a compact low dimensional vector.

- PCA feature reduction technique improves the efficiency of our approaches on action recognition. The feature dimensions can be reduced by 1/64 while preserving high precision and speeding up matching process with up to 8x.

- For multi-class classification, our experiments show that FV performs very well even with linear SVM, while BoF can only work with non-linear SVM.

- For the overall proposed method, we achieve 90.8% on precision and the speed of 56.6 frames per seconds by applying dense sampling. In practice, it only takes few seconds to recognize an action.

Chapter 6

Summary and Conclusion

This thesis describes a real-time action recognition system based on applying a GBH descriptor over divided video sequences. We first applied HOG people detection to detect humans, and an ST descriptor called the GBH descriptor was then employed. After that, we achieved feature representation by using the BoF approach and the FV; finally, both non-linear and linear SVM classifications were evaluated according to the representation methods. Our experiments showed that performance was preserved even when the spatial resolution was greatly reduced. Compared to existing methods, the major strengths of our method are its very high computational efficiency, and its potential for embedded applications.

6.1 Summary

Developing an action recognition system normally involves taking three steps: extracting ST features, representing them, and classifying the video based on the feature representations. We have proposed solutions to address the recognition challenges.

Because we divided each video into several sequences, and we detected active people at the beginning of the process, dense sampling could be used for the preprocessed sequences of the feature descriptor. This captured enough information and avoided a high dimension at the same time. We further employed a new local ST descriptor, called the GBH descriptor. This descriptor also uses a histogram of orientation method, as in HOG descriptors; however, in order to emphasize moving edge boundaries, time-derivatives of image gradients were used instead of image gradients. In-depth experimental evaluations showed that this simple gradient subtraction works well when the camera and background are largely static, and it also performs well on realistic datasets that contain a lot of camera motion and dynamic backgrounds. The important achievement of the GBH descriptor is its high efficiency, which makes it appropriate for use in surveillance systems having multiple static cameras.

We proposed a new local 3D descriptor for feature representation based on histograms of oriented spatial-temporal gradients for encoding both local static appearance and motion information. It significantly outperformed the popular HOG descriptor with high computational efficiency. We then applied a PCA feature reduction technique to improve the efficiency of our approach on action recognition. Our experiments showed that a feature dimension can be reduced by 63/64 through PCA, while preserving high accuracy, and speeding up matching processing with low dimension features. Those features could substantially speed up the BoF processing. We also evaluated the performance of the FV. Our preliminary experiments on the GBH descriptor with the FV showed that even with the linear classification, we could achieve higher efficiency.

With respect to the features produced by the BoF approach with the FV, we enforced a non-linear multi-class SVM and linear multi-class SVM for classification purposes. When combining FV with linear SVM, the precision was high enough and the efficiency

is quiet competitive.

6.2 Future Work

The GBH can only encode moving human gradients by using human detection and human tracking and by removing background information. For more improvements, a robust background and foreground subtraction algorithm and camera motion removal techniques should be taken into consideration.

Appendices

References

[1] Saad Ali and Mubarak Shah. "Human action recognition in videos using kinematic features and multiple instance learning". In: *IEEE transactions on pattern analysis and machine intelligence* 32.2 (2010), pp. 288–303.

[2] David Arthur and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding". In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics. 2007, pp. 1027–1035.

[3] Moez Baccouche et al. "Sequential deep learning for human action recognition". In: *International Workshop on Human Behavior Understanding*. Springer. 2011, pp. 29–39.

[4] Moshe Blank et al. "Actions as space-time shapes". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. IEEE. 2005, pp. 1395–1402.

[5] Aaron F. Bobick and James W. Davis. "The recognition of human movement using temporal templates". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.3 (2001), pp. 257–267.

[6] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: a library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27.

[7] Gabriella Csurka et al. "Visual categorization with bags of keypoints". In: *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. 1-22. Prague. 2004, pp. 1–2.

[8] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 886–893.

[9] Navneet Dalal, Bill Triggs, and Cordelia Schmid. "Human detection using oriented histograms of flow and appearance". In: *European conference on computer vision*. Springer. 2006, pp. 428–441.

[10] Piotr Dollár et al. "Behavior recognition via sparse spatio-temporal features". In: *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. IEEE. 2005, pp. 65–72.

[11] Alexei A Efros et al. "Recognizing action at a distance". In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE. 2003, pp. 726–733.

[12] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. "Learning precise timing with LSTM recurrent networks". In: *Journal of machine learning research* 3.Aug (2002), pp. 115–143.

[13] Raffay Hamid et al. "Detection and explanation of anomalous activities: Representing activities as bags of event n-grams". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 1031–1038.

[14] Tommi S Jaakkola, David Haussler, et al. "Exploiting generative models in discriminative classifiers". In: *Advances in neural information processing systems* (1999), pp. 487–493.

[15] Mihir Jain, Herve Jegou, and Patrick Bouthemy. "Better exploiting motion for better action recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2555–2562.

[16] Herve Jegou et al. "Aggregating local image descriptors into compact codes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.9 (2012), pp. 1704–1716.

[17] Gunnar Johansson. "Visual perception of biological motion and a model for its analysis". In: *Perception & psychophysics* 14.2 (1973), pp. 201–211.

[18] Yan Ke and Rahul Sukthankar. "PCA-SIFT: A more distinctive representation for local image descriptors". In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2004, pp. II–506.

[19] Yan Ke, Rahul Sukthankar, and Martial Hebert. "Efficient visual event detection using volumetric features". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 1. IEEE. 2005, pp. 166–173.

[20] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. "A spatio-temporal descriptor based on 3d-gradients". In: *BMVC 2008-19th British Machine Vision Conference*. British Machine Vision Association. 2008, pp. 275–1.

[21] Ivan Laptev and Tony Lindeberg. "Local descriptors for spatio-temporal recognition". In: *Spatial Coherence for Visual Motion Analysis*. Springer, 2006, pp. 91–103.

[22] Ivan Laptev and Tony Lindeberg. "Space-time Interest Points". In: *IN ICCV*. 2003, pp. 432–439.

[23] Ivan Laptev et al. "Learning realistic human actions from movies". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[24] Ivan Laptev et al. "Learning realistic human actions from movies". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[25] Ivan Laptev et al. "Learning realistic human actions from movies". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[26] David Marr and Herbert Keith Nishihara. "Representation and recognition of the spatial organization of three-dimensional shapes". In: *Proceedings of the Royal Society of London B: Biological Sciences* 200.1140 (1978), pp. 269–294.

[27] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. "Actions in context". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 2929–2936.

[28] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. "Trajectons: Action recognition through the motion analysis of tracked features". In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 514–521.

[29] Ross Messing, Chris Pal, and Henry Kautz. "Activity recognition using the velocity histories of tracked keypoints". In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 104–111.

[30] Eric Nowak, Frédéric Jurie, and Bill Triggs. "Sampling strategies for bag-of-features image classification". In: *European conference on computer vision*. Springer. 2006, pp. 490–503.

[31] Florent Perronnin and Christopher Dance. "Fisher kernels on visual vocabularies for image categorization". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–8.

[32] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. "Improving the fisher kernel for large-scale image classification". In: *European conference on computer vision*. Springer. 2010, pp. 143–156.

[33] Florent Perronnin et al. "Large-scale image retrieval with compressed fisher vectors". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 3384–3391.

[34] Ramprasad Polana and Randal Nelson. "Low level recognition of human motion (or how to get your man without finding his body parts)". In: *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*. IEEE. 1994, pp. 77–82.

[35] Deva Ramanan and David A Forsyth. "Automatic annotation of everyday movements". In: *Advances in neural information processing systems*. 2003, None.

[36] MD Rodriguez, J Ahmed, and M Shah. "Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*.

[37] Rasmus Rothe, Matthieu Guillaumin, and Luc Van Gool. "Non-maximum suppression for object detection by passing messages between windows". In: *Asian Conference on Computer Vision*. Springer. 2014, pp. 290–306.

[38] Jorge Sánchez et al. "Image classification with the fisher vector: Theory and practice". In: *International journal of computer vision* 105.3 (2013), pp. 222–245.

[39] Christian Schuldt, Ivan Laptev, and Barbara Caputo. "Recognizing human actions: a local SVM approach". In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 3. IEEE. 2004, pp. 32–36.

[40] Paul Scovanner, Saad Ali, and Mubarak Shah. "A 3-dimensional sift descriptor and its application to action recognition". In: *Proceedings of the 15th ACM international conference on Multimedia*. ACM. 2007, pp. 357–360.

[41] Feng Shi, Robert Laganiere, and Emil Petriu. "Gradient boundary histograms for action recognition". In: *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE. 2015, pp. 1107–1114.

[42] Andrea Vedaldi and Brian Fulkerson. "VLFeat: An open and portable library of computer vision algorithms". In: *Proceedings of the 18th ACM international conference on Multimedia*. ACM. 2010, pp. 1469–1472.

[43] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. "Differential recurrent neural networks for action recognition". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 4041–4049.

[44] OR Vincent and Olusegun Folorunso. "A descriptive algorithm for sobel image edge detection". In: *Proceedings of Informing Science & IT Education Conference (InSITE)*. Vol. 40. 2009, pp. 97–107.

[45] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features". In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2001, pp. I–511.

[46] Heng Wang et al. "A robust and efficient video representation for action recognition". In: *International Journal of Computer Vision* (2015), pp. 1–20.

[47] Heng Wang et al. "Action recognition by dense trajectories". In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 3169–3176.

[48] Heng Wang et al. "Evaluation of local spatio-temporal features for action recognition". In: *BMVC 2009-British Machine Vision Conference*. BMVA Press. 2009, pp. 124–1.

[49] Liang Wang and David Suter. "Informative shape representations for human action recognition". In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 2. IEEE. 2006, pp. 1266–1269.

[50] Daniel Weinland and Edmond Boyer. "Action recognition using exemplar-based embedding". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–7.

[51] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. "An efficient dense and scale-invariant spatio-temporal interest point detector". In: *European conference on computer vision*. Springer. 2008, pp. 650–663.

[52] Junji Yamato, Jun Ohya, and Kenichiro Ishii. "Recognizing human action in time-sequential images using hidden markov model". In: *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*. IEEE. 1992, pp. 379–385.

[53] Pingkun Yan, Saad M Khan, and Mubarak Shah. "Learning 4D action feature models for arbitrary view action recognition". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–7.

[54] Alper Yilmaz and Mubarak Shah. “A differential geometric approach to representing the human actions”. In: *Computer Vision and Image Understanding* 109.3 (2008), pp. 335–351.